©Copyright 2018

Eric Sander Lavine

The Evolution and Dynamics of Magnetized Plasma Jets in the MoCHI.LabJet Experiment

Eric Sander Lavine

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:
Setthivoine You, Chair
Raymond Golingo
Uri Shumlak

Program Authorized to Offer Degree
Aeronautics & Astronautics

University of Washington

Abstract

The Evolution and Dynamics of Magnetized
Plasma Jets in the MoCHI.LabJet Experiment

Eric Sander Lavine

Chair of the Supervisory Committee:
Setthivoine You

Magnetized plasma jets are generally modeled as magnetic flux tubes filled with flowing plasma governed by magnetohydrodynamics (MHD). Recent theoretical work has outlined a more fundamental approach based on flux tubes of canonical vorticity, where canonical vorticity is defined as the circulation of a species' canonical momentum. This approach extends the concept of magnetic flux tube evolution to include the effects of finite particle momentum and enables visualization of the topology of plasma jets in regimes beyond MHD. Under the appropriate conditions, this framework suggests how to form and drive stable, collimated plasma jets with very long aspect ratios. To explore this possibility, a triple electrode planar plasma gun (MoCHI.LabJet) has been designed to produce helical shear flows inside a driven, magnetized plasma jet. This thesis presents the motivation, foundational theory, design, and initial results from this novel pulsed power experiment. The formation of long (1.1 m), collimated, stable jets with aspect ratios $\gtrsim 20{:}\,1$ was observed following an improvement to the insulation between the gun electrodes. Ion Doppler spectroscopy measurements of these jets suggest the presence of strong helical flows with sufficient shear to stabilize current driven kink-instabilities. Magnetic field measurements suggest an additional mechanism for enhanced stability by indicating that the magnetic field of the jet is qualitatively similar to a Taylor double helix. Strong flows, a plasma $\beta \approx 1$, and the unbounded nature of jets in the MoCHI experiment; however, are inconsistent with a Taylor state, suggesting a unique and heretofore unobserved plasma state that may be best described as a generalized, non-Taylor, driven equilibrium.

Acknowledgements

It is difficult to express the gratitude I feel toward everyone who has helped me on this odyssey of an endeavor, but I will certainly try. First, I would like to thank Professor Setthivoine You, who took me under his wing as an untested graduate student with no prior training in plasma physics or experimental research and entrusted me with a vast amount of freedom and responsibility on projects ranging from the theoretical to the development of hardware required to obtain the remarkable results presented in this thesis. It is under his incredible vision that this lab was realized and I feel truly privileged to have been a part of it. I would also like to thank my colleagues and fellow graduate students: Alexander Card, Evan Carroll, Iman Datta, Manuel Azuara-Rosales, Morgan Quinley, Keon Vereen, Yu Kamikowa, Yu Takagaki and, in particular, Jens von der Linden. Jens was a pillar of support for both myself and the MoCHI lab and I am grateful for the discussions we have had and for his friendship. I am indebted to the faculty of the Aero-Astro department: Prof. Carl Knowlen, Prof. Brian Nelson, Prof. Tom Jarboe, Prof. Uri Shumlak, Prof. Eric Meier, and especially Prof. Raymond Golingo who stepped up to help me out when the going got tough and provided much needed encouragement and advice.

I would never have arrived at this point without the unconditional love and support of my parents, Bruce Lavine and Marjolijn Plomp. When feeling abashed over sheer magnitude of all they have done for me throughout my life and over the last seven years, my father once told me "The harpoonist doesn't row". I am so grateful for all the rowing they have done to get me here. Hopefully this thesis can represent the successful capture of the great white whale we were tracking down all these years. I am also thankful for my sister Anika, who remains the kindest person I know and was somehow able to put up with me growing up. I couldn't be prouder to watch her overcome adversity and blaze her own path through academia with aplomb. I am also grateful to be able to share this accomplishment with my grandfather Richard Lavine, whose passion for science rubbed off on me at an early age and likely set me on this path.

Despite stealing their daughter away, I am also privileged to have had the unconditional support of my in-laws Gregg Greenberg and Jamie Goren. They welcomed me into their family as one of their own and understood to tread lightly when asking me "how much longer until you're done?". I look forward to spending more time together in the upcoming years. And finally, most of all, I would like to thank my best friend and wife Jaclyn Lavine. Words fall short to express the gratitude I feel for her unwavering support and the sacrifices she has made on my behalf. Of all

the decisions I have made and things I have accomplished, marrying her continues to be the smartest and most rewarding.

Table of Contents

1. Introduction **1**
1.1 The plasma universe 1
1.2 The many descriptions of plasmas 3
1.2.1 A kinetic description of plasmas 3
1.2.2 A multifluid description of plasmas 5
1.2.3 A single-fluid description of plasmas 6
1.3 Magnetohydrodynamic (MHD) theory 8
1.3.1 Frozen-in flux condition 8
1.3.2 Magnetic helicity 9
1.3.3 Constrained relaxation and force-free states 11
1.4 Context: astrophysical and laboratory plasmas 16
1.4.1 Astrophysical jets 16
1.4.2 Coronal loops, magnetic clouds, and the coronal heating problem 17
1.4.3 Merging experiments 19

2. A Unified Model 20
2.1 Introduction 20
2.2 Definitions and equations 20
2.3 Canonical flux tubes and their helicity 22

3. The Topology of Canonical Flux Tubes in Flared Jet Geometry 28
3.1 Introduction 28
3.2 Visualization of canonical flux tubes 29
3.3 Kinematic evolution of flared plasma jet 37
3.3.1 Uniform and diffuse current profiles 38
3.3.2 Diffuse-core-with-skin and skin-only current profiles 45
3.4 Kinematic evolution of relative canonical helicity 50
3.5 Summary 54

4. The MoCHI.LabJet Experiment **56**
4.1 Introduction 56

4.2 Laboratory astrophysics 56
4.2.1 Scaling relations 58
4.3 Triple electrode planar plasma gun 60
4.4 Vacuum and fueling system 62
4.5 Power supplies 64
4.6 Control and data acquisition 66
4.7 Diagnostics 68
4.8 First plasmas 71

5. Ion Doppler Spectroscopy Measurements on the MoCHI.LabJet Experiment 84
5.1 Introduction 84
5.2 Hardware 86
5.2.1 Calibration 86
5.3 Line integrated plasma parameters 89
5.3.1 Hydrogen plasmas - H_α emission 90
5.3.2 Hydrogen plasmas - CII impurity emission 96
5.3.3 Nitrogen plasmas - NII emission 100
5.4 Estimating local plasma velocities 106
5.4.1 Double shell model 106
5.4.2 Abel inversion 114
5.5 Discussion 120

6. Magnetic Field Measurements on the MoCHI.LabJet Experiment 123
6.1 Theory and calibration 123
6.2 Experimental data 128
6.3 Magnetic field cross section from a single linear $\dot{B}$ probe array 136
6.4 Comparison to helical Taylor state 145
6.5 Discussion 151

7. Conclusion 153

8. Suggestions for Future Work 159

Bibliography **161**

Appendix A: Fast Ionization Gauge for Neutral Gas Density Measurements **166**
A.1 Operation 166
A.2 Principle 166
A.3 Electrical design 168
A.4 Vacuum design 170
A.5 Calibration 170
A.6 Characterization of fast gas valves 173

Appendix B: Development of Custom Multichannel Optical Fiber Bundle **177**

Appendix C: Flux Tube Visualization Codes **183**

Appendix D: Stability space codes **193**

Appendix E: IDS Codes **195**

Appendix F: Magnetic Field Codes **215**

Appendix G: Current and Voltage WaveForm Code **293**

Appendix H: FIG Codes **298**

LIST OF FIGURES

Figure Number **Page**

1.1 Various naturally occurring and laboratory plasmas and their position in density-temperature space. For an ionized gas to be considered a plasma, one key requirement is that the characteristic dimensions must be much larger than the Debye length $\lambda_D = \sqrt{(\epsilon_0 k_B T_e / n_0 q_e^2)}$ which is the distance at which charge is shielded within a plasma. For most terrestrial plasmas, the Debye length $< 1\text{mm}$. For astrophysical plasmas however, the Debye length can be tens of meters. Image Credits: Interstellar Gas, NASA/Hubble; Nebulae, NASA/ESA/J. Hester and A. Loll (Arizona State University); Solar Corona, ESA & NASA/SOHO; Lightning, NOAA; Magnetic Confinement Fusion, MAST tokamak, Culham centre for fusion energy; Solar Core, SOHO/EIT consortium; Inertial confinement Fusion, Lawrence Livermore National Laboratory. YSO jets, J. Morse/STSci, and NASA/ESA. **2**

1.2 Relationship between individual particle trajectories and the electromagnetic fields. If all fields and particle trajectories are known, the evolution of the plasma can be determined exactly. **4**

1.3 Representation of gauge ambiguity for two intertwined magnetic flux tubes within a bounding volume V (grey box). If magnetic flux leaves the bounding volume, the magnetic helicity integral cannot count the number of linkages, writhes and twists of the two flux tubes external to the volume. Adapted from [5]. **10**

1.4 Magnetic field line topology of the Lundquist solution in an infinite cylinder (left) and cylindrical spheromak in a flux conserver with an aspect ratio $\mathrm{L/a} = 1$ (right). The spheromak fields form toroidal helices around closed magnetic flux surfaces and are purely poloidal $(\hat{\mathrm{r}}, \hat{\mathrm{z}})$ at the boundaries (black lines). **14**

1.5 (a) The determinant of equation 1.28 evaluated for an aspect ratio of 10 and $\mathrm{m} = 1$. The first two physical roots appear at $\lambda\mathrm{a} = 3.15$ and $\lambda\mathrm{a} = 3.27$. (b) Dependence of $\lambda\mathrm{a}$ on the aspect ratio of the flux conserving volume for the first two solutions with $\mathrm{m} = 0$ and $\mathrm{m} = 1$ symmetry. The minimum energy state (smallest λ) is axisymmetric for $\mathrm{L/a} \lesssim 1.67$ and nonaxisymmetric for $\mathrm{L/a} \gtrsim 1.67$. Adapted from [17]. **15**

1.6 Isosurfaces of the lowest order $m = 1$ solution of a constant λ force-free state for $L/R = 10$. Field streamlines (red) have been included in the plot of isosurfaces for B_z. The structure is fundamentally helical, decaying in strength near the end wall at $z = 0$ and $z = L$..... **16**

3.1 Example flux coordinate directions for (a) a magnetic flux tube ψ, (b) a flow vorticity flux tube $\mathcal{F}_\sigma$ of species σ, and (c) canonical vorticity flux tube Ψ_σ for the species σ. The directions are determined by the scalar flux functions (Table 1). The canonical vorticity flux coordinates in panel (c) are the weighted sum of the magnetic and flow vorticity components of panels (a) and (b). **30**

3.2 Shape of the axial current density j_z as a function of normalized radius η for (a) the uniform current profile, (b) the diffuse core current profile, (c) the diffuse-core-with-skin, and (d) skin-only current profile. The splines used in (c) and (d) are fifth-order interpolating polynomials set to match the value and slope of the core current at $\eta = 0.6$ with a maximum at $\eta = 0.8$ and going smoothly to zero at $\eta = 1$. These profiles are scaled by a multiplicative constant $j_0(z)$ to retrieve the total jet current I_{jet} when integrated over the cross-sectional area at any axial location (Equation 3.4)..... **32**

3.3 Five equivalent interpretations of flux tubes from the most familiar to the most abstract: (a) a current-carrying magnetic flux tube ψ_{jet} with uniform current profile and bulk flow velocity $\overline{\mathbf{U}}$ for $l_0^{-1}\lambda \approx 2.5$ and $|\overline{\mathbf{u}}_\mathbf{e}| = 3$; (b) ion and electron fluid velocity flux tubes ψ_σ with flows restricted to magnetic flux surfaces; (c) ion and electron fluid vorticity flux tubes $\mathcal{F}_\sigma$; (d) ion and electron canonical momentum flux tubes $\mathcal{P}_\sigma$; (e) ion and electron canonical vorticity flux tubes Ψ_σ. Here, the electron flux tube is coincident with the magnetic flux tube (light gray in panels a, b, and e) because of negligible electron mass while the ion flux tube of interest is highlighted (dark gray in panel e).. ..... **37**

3.4 Flared magnetic flux tubes with uniform and diffuse core current profiles for sub-Alfvénic ($|\bar{\mathbf{u}}_e| = 2/3$) and super-Alfvénic ($|\bar{\mathbf{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. Thin, solid lines are magnetic field streamlines ($\bar{\mathbf{B}}$), thin dashed lines are current density streamlines ($\bar{\mathbf{j}}$), and bold dashed lines are bulk flow velocity streamlines ($\bar{\mathbf{U}}$). Arrows on the streamline indicate the direction of the vector field. Streamlines on both the outer magnetic flux surface (ψ_{jet}) and an inner magnetic flux surface (ψ_{inner}) are shown; however, some streamlines on the outer magnetic flux surface have been truncated for visual clarity. Figure 5 shows the same systems from a canonical flux tube point-of-view..... **41**

3.5 Canonical vorticity flux tubes with uniform and diffuse core current profiles for sub-Alfvénic ($|\bar{\mathbf{u}}_e| = 2/3$) and super-Alfvénic ($|\bar{\mathbf{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. These canonical flux tubes are equivalent to the corresponding magnetic flux tubes of Fig. 4. Ion canonical vorticity streamlines ($\bar{\boldsymbol{\Omega}}_\mathbf{i}$) are drawn as bold, dashed lines along two ion canonical vorticity flux tubes (Ψ_i and $\Psi_{i,inner}$), while electron canonical vorticity ($\bar{\boldsymbol{\Omega}}_\mathbf{e}$) streamlines (geometrically equivalent to magnetic field streamlines), are depicted as a thin, solid line along an electron canonical flux tube ($\Psi_e \Leftrightarrow \psi_{jet}$, light gray tube). Some streamlines on the outer ion canonical vorticity flux tube (Ψ_i, dark gray tubes) have been truncated for visual clarity. As current is increased, ion canonical flux tubes depart from magnetic flux tube surfaces and become collimated. **42**

3.6 Dependence of ion canonical vorticity flux surface Ψ_i (dashed lines) on total jet current $\bar{\lambda}$ (panels a, c) and on electron fluid velocity $|\bar{\mathbf{u}}_e|$ (panels b, d) for uniform (panels a, b) and diffuse core (panels c, d) currents. For both current profiles, full collimation occurs closer to the base of the flux tube with increasing electron velocity $|\bar{\mathbf{u}}_e|$. A prominent bulge in the canonical flux tube occurs at low $\bar{\lambda}\sim0.5$ and high $|\bar{\mathbf{u}}_e| = 10$ for a diffuse core current profile (panel c). Plots of current profile $\bar{j}_z(\bar{r})$ are shown at the base of the jet (lightest gray) while representative poloidal flow vorticity profiles $\bar{\omega}_{ir}(\bar{r})$ (dark gray) and $\bar{\omega}_{iz}(\bar{r})$ (light gray) are drawn at $\bar{z} = 15$. Contours of constant magnetic flux ψ (solid lines) do not change with $\bar{\lambda}$ nor $|\bar{\mathbf{u}}_\mathbf{e}|$ and are presented to reveal passage of the canonical flux surface through different regions of the current profile. **45**

3.7 **(Left)** Bulged ion canonical vorticity flux tube Ψ_i (dark gray) for a diffuse core current jet with low current ($\bar{\lambda} = 0.175$) and a large electron velocity ($|\bar{\mathbf{u}}_e| = 40$). Equation (5) shows that density gradients can be normal to the dark gray canonical Ψ_i surfaces rather than the light gray magnetic flux surfaces ψ suggesting that ion line emission reveals canonical flux tubes as opposed to magnetic flux tubes. **(Right)** M2-9 Butterfly nebula (NASA HST STIS/CCD – MIRVIS), a bipolar planetary nebula with distinctive bulging that resembles the ion canonical flux tube in the left panel. Transitions in the geometry of the jet (i.e. flaring to collimation) may therefore also reflect changes in the relative strength of the fluid flow vorticity flux to the magnetic flux. Image credit: Bruce Balick (University of Washington), Vincent Icke (Leiden University), Garret Mellema (Stockholm University), and NASA **44**

3.8 Flared magnetic flux tubes with diffuse-core-with-skin and skin-only current profiles for sub-Alfvénic ($|\bar{\mathbf{u}}_e| = 2/3$) and super-Alfvénic ($|\bar{\mathbf{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. Thin solid lines are magnetic field streamlines ($\bar{\mathbf{B}}$), thin dashed lines are current density streamlines ($\bar{\mathbf{J}}$), and bold dashed lines are bulk flow velocity streamlines ($\bar{\mathbf{U}}$). Arrows on the streamline indicate the direction of the vector field. Streamlines on both the outer magnetic flux surface (ψ_{jet}) and an inner magnetic flux surface (ψ_{inner}) are shown; however, some streamlines on the outer magnetic flux surface have been truncated for visual clarity. Figure 5 shows the same systems from a canonical flux tube point-of-view. **47**

3.9 Canonical vorticity flux tubes with diffuse-core-with-skin and skin-only current profiles for sub-Alfvénic ($|\bar{\mathbf{u}}_e| = 2/3$) and super-Alfvénic ($|\bar{\mathbf{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. These canonical flux tubes are equivalent to the corresponding magnetic flux tubes of Figure 8. Ion canonical vorticity streamlines ($\bar{\boldsymbol{\Omega}}_i$) are drawn as bold, dashed lines along two ion canonical vorticity flux tubes (Ψ_i and $\Psi_{i,inner}$), while electron canonical momentum ($\bar{\boldsymbol{\Omega}}_e$) streamlines (geometrically equivalent to magnetic field streamlines), are depicted as a thin, solid line along an electron canonical flux tube ($\Psi_e \Leftrightarrow \psi_{jet}$, light gray tube). Some streamlines on the outer ion canonical vorticity flux tube (Ψ_i, dark gray tubes) have been truncated for visual clarity. As the current is increased, ion canonical flux tubes depart from magnetic flux tube surfaces and begin to collimate. For the skin current (e-h), collimation never completes due to the absence of poloidal flow vorticity flux in the core of the jet but the flaring is less pronounced than the flare of the magnetic flux tube. **48**

3.10 Dependence of ion canonical vorticity flux surface Ψ_i on total jet current $\bar{\lambda}$ (panels a, c) and on electron fluid velocity $|\bar{\mathbf{u}}_\mathbf{e}|$ (panels b, d) for diffuse-core-with-skin (panels a, b) and skin-only (panels c, d) currents. For both current profiles, canonical flux surfaces separate closer to the base of the jet with increasing jet current $\bar{\lambda}$ and electron velocity $|\bar{\mathbf{u}}_\mathbf{e}|$; however, for the skin-only current, collimation does not occur due to the absence of poloidal flow vorticity flux in the core of the jet. Plots of the current profile $\bar{j}_z(\bar{r})$ are shown at the base of the jet (lightest gray) while representative poloidal flow vorticity profiles $\bar{\omega}_{ir}(\bar{r})$ (dark gray) and $\bar{\omega}_{iz}(\bar{r})$ (light gray) are drawn at $\bar{z} = 15$. Contours of constant magnetic flux ψ (solid lines) do not change with $\bar{\lambda}$ nor $|\bar{\mathbf{u}}_\mathbf{e}|$ and are presented to reveal passage of the canonical flux surface through different regions of the current profile **49**

3.11 Kinematic evolution of the weighted normalized ion relative canonical helicity components for select current profiles and electron flow velocities. For all the cases considered in Section 4 cross and magnetic helicity components dominate the total canonical helicity content of the jet. Ion kinetic helicity increases at moderate values of $\bar{\lambda}$ but levels off or even decreases as the ion flows unwind at high $\bar{\lambda}$ (see the text, Figure 13). For the diffuse core current profile (panels c, d), a negative dip in the ion flow helicity is observed at moderate $\bar{\lambda}$ as a consequence of a changing $\bar{\omega}_{i\theta}(r)$ profile..... **52**

3.12 Kinematic evolution of the normalized ion relative canonical helicity components for a uniform current profile with a normalized electron flow velocity $|\bar{\mathbf{u}}_\mathbf{e}| = 40$. In this case, the ion kinetic helicity dominates at moderate $0.2 \lesssim \bar{\lambda} \lesssim 10$, with cross-helicity dominating at larger $\bar{\lambda}$..... **53**

3.13 Ion flow tubes for the case of a uniform current profile with increasing jet current $\bar{\lambda}$. Ion flow streamlines (dashed line) twist as $\bar{\lambda}$ is increased (panel a to b) then unwind at even larger $\bar{\lambda}$ (panel b to c) to compensate for increasing electron twist (solid line) in carrying the poloidal current. This behavior explains the reduction in ion flow helicity observed at large $\bar{\lambda}$ values for current profiles that are more distributed throughout the cross-section of the jet (Figure 3.11). **53**

4.1 Concept behind the Labjet configuration of the mochi experiment: The setup replicates the basic configuration of an accretion disk rotating around a magnetized central engine with three concentric electrodes threaded by a poloidal magnetic field. The inner electrode (red) is designed to drive the core of the plasma jet and the middle electrode (yellow) to drive the outer skin of the jet, and therefore provide some degree of control on the jet current and flow profile. The mass sources are four azimuthally symmetric gas slits at the electrodes, allowing for unhindered azimuthal rotation. **61**

4.2 Full LabJet experimental setup with the spiderleg gun mounted opposite to the LabJet gun on the 1.4m diameter spherical vacuum chamber. Numerous viewports provide diagnostic and spectroscopic access and are arranged to point both radially inward as well as orthogonal to the $\hat{z}$ axis..... **63**

4.3 Left) Cutaway of custom fast gas valve. Right) Complete array of 10 custom fast gas valves, each powered and triggered independently via a 10-channel variable voltage power supply. Reproduced from [86]..... **64**

4.4 left) Circuit diagram of PSU2 and PSU3, which control the middle and inner electrodes respectively. More detail regarding the design, fabrication, and optically isolated control of the power supplies can be found in reference 78. Right) Picture of PSU 2 showing the 5 capacitors, internal circuitry and custom brass ignitron tower. **65**

4.5 Simplified circuit diagram of the 10-channel, variable voltage power supply for control of the custom fast gas valves. Each channel is can be charged to a different voltage by adjusting the position of the variable resistor terminal. Firing times are controlled by 10 optotrigger circuits. **66**

4.6 a) End on view of one possible magnetic probe array configuration, two arrays have a 7.5 cm dogleg and can be rotated into and out of the plasma column. b) Three arrays installed at different axial positions. c) Layout of Bdot probe array which consists of a high resolution central section with 25 clusters of 3 orthogonal inductor chips bordered by two low resolution sections with 10 clusters. **69**

4.7 a) Horiba 1000M f/8 spectrometer. b) Matching and collecting optics (back to back Nikon 35, 105mm lenses and a PI-MAX3 ICCD camera. c) CCD view of optical fiber bundle backlit with hydrogen reference lamp, individual bundles have been labeled. d) Custom 10m long optical fiber bundle, consisting of a 3×32 channel v-groove chip clamp, 12-way fiber breakout, and 12×8 channel v-groove clamps with laser diode collimating lenses. **71**

4.8 a)Time evolution of flared plasma jet constructed from shots 3080-3091 by stepping camera acquisition times by $2\mu s$. Shots were conducted using diatomic hydrogen with the middle electrode only at -2.5kV. Arcing across insulating gap appears to disrupt plasma jet development b) Current traces for shots 3080 – 3091 reveal total jets currents of ~ 20kA. c) Voltage traces for shots 3080-3091, the initial drop at $t = 0$ reveals when the gun ignitron is fired, breakdown occurs shortly afterward at $t \approx 1\mu s$. Figures adapted from [78]..... **72**

4.9 a) Azimuthally symmetric breakdown during the 'bundt cake' formation phase when operating the middle electrode only. Gas injection is provided by a custom valve to the middle gas injection slit and four parker valves to the outer-inner gas injection slit. Images reveal that the outer footpoints do not terminate at the outer-inner gas injection slit as expected. b) Similar breakdown occurs when no gas is injected into the outer-inner slit. These observations suggest insufficient gas is being provided to the outer-inner slit. **73**

4.10 ICCD images of various plasma shots (inner electrode only) taken at the same time and jet current for different vacuum magnetic field strengths. The development of a current driven kink instability is observed to depend on the strength of the bias magnetic flux ψ_{gun}. The kink occurs more rapidly at lower bias magnetic fields (larger $\lambda = \mu_0 I_{gun}/\psi_{gun}$), in qualitative agreement with the Kruskal-Shafranov instability condition (equation 4.4). **74**

4.11 Typical operation sequence for first operational campaign with Teflon insulator between middle and outer electrodes. a) Vacuum bias magnetic field is applied. b) Neutral gas is injected by 10 custom fast gas valves to form an azimuthally symmetric plume in front of the electrodes. c) Ignitrons are fired and breakdown occurs along magnetic field lines linking the inner and outer-outer gas slits (red core-current) and middle and outer-inner gas slits (orange skin-current). This formation phase is termed the 'Bundt cake' due to the physical resemblance between the nested arches and a Bundt cake. An azimuthal flow due to $\vec{E} \times \vec{B}$ forces develops. d) The jet zippers up along the central axis, driving axial shear flows and convecting plasma from the base of the jet. e) Convection of toroidal magnetic flux leads to collimation of the plasma column. f) Jet reaches critical aspect ratio for a given jet current and bias flux, resulting in a kink instability which breaks the jet. g) Initial jet propagates as detached plasmoid and a secondary diffuse jet forms, bordered by bright arches over the Teflon insulator. **75**

4.12 Characteristic discharge (#6580) during the first operational campaign of the LabJet experiment. ($V_{core} = -4.6\text{kV}, V_{skin} = -4\text{kV}, \psi_{gun} = 3.4\text{mWb}$) a) Select frames from high speed video taken with the Kirana camera. The red dot indicates the path of the He-Ne interferometer providing electron density measurements and each frame number corresponds to the labeled red dots in panel c. b) Gun currents measured using calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. Panels (a) and (c) reproduced from [86]..... **77**

4.13 Evolution of the primary jet. a) High speed camera images taken with Kirana at 2Mfps, light emission is used to calculate the length of the plasma jet (frame 7). b) Axial velocity of 70.6 km/s is calculated by plotting the jet length as a function of time. c) Evolution of the jet through stability space indicates that jet goes unstable at the predicted Kruskal-Shafranov instability limit..... **78**

4.14 Estimate of temperature evolution within the primary jet assuming Spitzer resistivity. The blue line shows the core-determined temperature using the inner electrode voltage and current waveforms, while the green line shows the skin determined temperature. **78**

4.15 Typical operation sequence for second and third operational campaigns with alumina insert between middle and outer electrodes. a) Vacuum bias magnetic field is applied. b) Neutral gas is injected by 10 custom fast gas valves to form an azimuthally symmetric plume in front of the electrodes. c) Ignitrons are fired and breakdown occurs along magnetic field lines linking the inner and outer-outer gas slits (red-core current) and middle and outer-inner gas slits (orange-skin current). This formation phase is termed the 'Bundt cake' due to the physical resemblance between the nested arches and a Bundt cake. An azimuthal flow due to $\vec{E} \times \vec{B}$ forces develops. d) The jet zippers up along the central axis, driving axial shear flows and convecting plasma from the base of the jet. e) Convection of toroidal magnetic flux leads to collimation of the plasma column. f) Jet reaches critical aspect ratio for a given jet current and bias flux, resulting in a kink instability which breaks the jet. g) Initial jet propagates as detached plasmoid and a secondary diffuse jet forms. h) Secondary diffuse jet lengthens into the plasmoid. i) Secondary jet collimates into long, stable jet. **80**

4.16 Characteristic discharge for the second operational campaign. a) High speed camera images taken with a Kirana camera at 1Mfps. The red dot indicates the path of the He-Ne interferometer and each frame number corresponds to the labeled red dots in panel c. b) Gun currents measured using calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density shows a spike for the initial jet and gradual rise during the formation of the long stable jet. d) Jet evolution through classical stability space. The initial jet lengthens and goes kink unstable near the Kruskal-Shafranov instability limit. The secondary diffuse plume forms in stable space and crosses into unstable space as the current increases. The long jet remains stable with a fixed length (1.1m) for tens of μs despite occupying classically unstable space **81**

4.17 Characteristic discharge for the third operational campaign. a) High speed camera images taken with a Kirana camera at 1Mfps. The red dot indicates the path of the He-Ne interferometer and each frame number corresponds to the labeled red dots in panel c. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density shows a spike as the detached plasmoid propagates across the chamber and rise when the bowed long stable jet intercepts the laser beam. d) Jet evolution through classical stability space. The initial jet lengthens and goes kink unstable at the Kruskal-Shafranov instability limit. The secondary diffuse plume forms in stable space and crosses into unstable space as the current increases. The long jet remains stable with a fixed length (1.1m) for tens of μs despite occupying classically unstable space. **82**

4.18 Temperature estimate from gun current and voltage waveforms assuming Spitzer resistivity. The waveforms are noisy during stage I and the start of stage II but are smooth during stage III, with a total resistance of $\sim 50 \mathrm{m}\Omega$. Temperature estimates account for inductive impedance; however, this contribution is an order of magnitude less than the resistive impedance. At the start of stage III, the core and skin temperatures are ~ 5 eV with the skin increasing rapidly to > 20 eV at $t \sim 60 \mu s$ and the skin increasing more gradually towards the end of the discharge. **83**

4.19 High speed camera images of several shots that exhibit a distinct helical structure. A similar structure is observed on most shots; however, it is often obscured due to saturation of the Kirana camera or, for the case of highly collimated jets, the helical winding is so tight that it is difficult to discern. **83**

5.1 Theoretical dispersion curve (a) and Doppler resolution (b) for various gratings. The value $\alpha_0 = 5.25$ in equation 5.4 was found to provide the best fit to measurements taken with a spectral lamp of known doublets. For H_α emission at 656.3nm, a 2400 l/mm grating provides a Doppler resolution of ~ 1.6 (km/s)/px. For CII emission at 723.64nm, the same grating provides a resolution of ~ 1.25 (km/s)/px. **87**

5.2 Reference images taken at various wavelengths. Rectangles indicate the binning volume in the vertical (spatial) direction. a) Reference image taken using a Hydrogen spectral lamp at 656.3nm. b) Reference image of CII impurities taken at a spectrometer central wavelength of 723.392 nm over a long exposure plasma shot. The two spectral lines are CII doublets at 723.132 nm and 723.642 nm (selected in red rectangles). c) Reference image of NII emission at 648.2 nm over a long exposure nitrogen plasma shot. Nearby NI emission can be seen to the left of the red rectangles (red shift in wavelength space) **89**

5.3 Comparison between shot data and the reference image for fiber 6-1 in pixel space (a) and wavelength space (b). Line integrated velocities are calculated from Δpx between the two centroids (related to $\Delta\lambda$ through the spectrometer dispersion relation) while temperatures are calculated from $\Delta\sigma$, based on the assumption that the standard deviation of the reference gaussian is representative of the instrument temperature. This instrument function (solid curve) is shifted and rescaled (dotted curve in panel (b)) for comparison with the shot data gaussian (dashed curve). **91**

5.4 Optical fiber bundles were divided into two measurement planes and viewing chords were determined by back-shining a laser through the fibers. a) The poloidal measurement plane for measurement of axial velocities. b) Toroidal measurement plane for measuring azimuthal velocities. **92**

5.5 Evolution of shot #7143. a) High speed camera images taken with the Kirana camera at 2Mfps. The red dot indicates the path of the interferometer. Panels 7 and 8 correspond to the timing of the first and second frames taken with the PI-MAX3 ICCD camera. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. d) ICCD images of the optical fiber bundle taken using the PI-MAX3 ICCD camera at 43μs and 58μs with an exposure time of 0.5μs. **94**

5.6 Line integrated velocities in the poloidal plane for shot #7143 using H_α emission lines. Increasing blue shift from fiber bundle 1 to fiber bundle 6 agrees with the predicted behavior; however, a negative ($\sim -10\mathrm{km/s}$) skew to the data is observed. Possible sources for the skewing of the data set include large radial outflows, bulk oscillation of the plasma column, or contributions from an azimuthal velocity. **94**

5.7 Line integrated velocities in the toroidal plane for shot #7143 using H_α emission lines. Indications of significant flow shear are observed for fiber bundles 7 and 8 but are not observed for the other fiber bundles which likely do not intercept the central plasma column. A negative skew to the data set of $\sim -10\mathrm{km/s}$ may be due to large radial outflows or bulk oscillation of the plasma column. **95**

5.8 Line integrated temperatures in the poloidal plane for shot #7143 using H_α emission lines. Significant heating is observed between $t = 43\mu s$ and $t = 58\mu s$. Sharp gradients in the temperature profile likely reflect whether the lines of sight are intercepting the central plasma column where temperatures are expected to be largest **97**

5.9 Line integrated temperatures in the toroidal plane for shot #7143 using H_α emission lines. The most pronounced heating is observed for fiber bundles 7 and 8 which exhibit a double peaked temperature profile. Less significant heating is observed for the remaining bundles which likely do not intercept the central plasma column..... **98**

5.10 Kirana camera images of a typical jet at $43\mu s$ and $59\mu s$ with the corresponding ICCD images of the optical fiber bundle measuring CII impurity emission at 723.64nm (red boxed). A second impurity line corresponding to CII at 723.13nm is also observed but was not used as it was slightly weaker. The full evolution of this shot, including gun current, voltage, line integrated electron density measurements and stability space analysis are presented in figure 4.16. ..... **101**

5.11 Line integrated velocities in the poloidal plane for shot #7150 using CII emission lines. Transitions from large red shift to blue shift with increasing bundle number follows the predicted behavior. Steep positive gradients for bundles 1 and 2 are observed at $t = 59\mu s$ possibly indicating deceleration due to pileup at the end wall. A large decrease in velocity between frames is also observed for bundle 6.. **102**

5.12 Line integrated velocities in the toroidal plane for shot #7150 using CII emission lines. At $t = 43\mu s$ the velocity profiles for bundles 7 and 8 are indicative of azimuthal rotation in the counter-clockwise direction based on a transition from blue shift to red shift with increasing view chord number. Similar profiles are observed for all bundles except bundle 10 at $t = 59\mu s$. This rotation direction is consistent with the predicted $\vec{E} \times \vec{B}$ direction of the LabJet gun.. ..... **103**

5.13 Evolution of shot #7240 with NII as the working gas. a) High speed camera images taken with the Kirana camera at 2Mfps. The red dot indicates the path of the interferometer. Panels 7 and 8 correspond to the timing of the first and second frames taken with the PI-MAX3 ICCD camera. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. d) ICCD images of the optical fiber bundle taken using the PI-MAX3 ICCD camera at $43\mu s$ and $58\mu s$ with an exposure time of $1\mu s$ at a central wavelength of 648.2nm... ..... **104**

5.14 Line integrated velocities in the poloidal plane for shot #7240 using NII emission lines. Transitions from red shift to blue shift with increasing bundle number at $t = 43\mu s$ follows the predicted behavior. At $t = 58\mu s$ bundles 1-5 are blue shifted by $\gtrsim 5$km/s possibly due to bulk oscillation of the central column and/or deceleration of the jet due to impact and reflection off the end wall. **104**

5.15 Line integrated velocities in the poloidal plane for shot #7240 using NII emission lines. Transitions from red shift to blue shift with increasing bundle number at $t = 43\mu s$ follows the predicted behavior. At $t = 58\mu s$ bundles 1-5 are blue shifted by $\gtrsim 5$km/s possibly due to bulk oscillation of the central column and/or deceleration of the jet due to impact and reflection off the end wall. **105**

5.16 Simple spectroscopy model which consists of a jet with two concentric cylindrical shells of equal thickness. The approximate center and radius of the jet in the toroidal midplane is calculated using gaussian fits to the spectral intensity as a function of view chord (Figures 5.17 and 5.18). The model has 8 free parameters which can be adjusted independently. The axial velocity profile was selected to be relatively flat in the core, going smoothly to zero at the jet radius. **107**

5.17 Calculation of the approximate jet center and radius for shot #7150 at $t = 43\mu s$. a) The centroid of gaussian fits to the total intensity (excluding bundles 10-12) allows for calculation of the approximate jet center position by minimizing the sum of the distances between a center point and the two view chords nearest to the centroid for each bundle. The approximate jet radius is calculated based on the average full width half max of the gaussian fits (excluding bundles 10-12) multiplied by the average spacing between view chords in the midplane (~ 2cm). b) High speed camera image of the jet with fiber bundle viewports marked. c) Reproduction of jet model based on calculated center position and jet radius. The clear cylinder is a projection of the middle electrode while the grey circle marks the extent of the outer electrode. **108**

5.18 Calculation of the approximate jet center and radius for shot #7150 at $t = 59\mu s$. a) The centroid of the gaussian fits allows for calculation of the approximate jet center position by minimizing the sum of the distances between a center point and the two view chords nearest to the centroid for each bundle. The approximate jet radius is calculated based on the average full width half max of the gaussian fits multiplied by the average spacing between view chords in the midplane (~ 2cm). b) High speed camera image of the jet with fiber bundle viewports marked. c) Reproduction of jet model based on calculated center position and jet radius. The clear cylinder is a projection of the middle electrode while the grey circle marks the extent of the outer electrode. **109**

5.19 Double shell model fits to the line integrated velocities in the poloidal plane for shot #7150 at $t = 43\mu s$. Bundles 3-5 have been ignored due to low signal to noise for the spectral intensities. **110**

5.20 Double shell model fits to the line integrated velocities in the toroidal plane for shot #7150 at $t = 43\mu s$. **111**

5.21 Double shell model fits to the line integrated velocities in the poloidal plane for shot #7150 at $t = 59\mu s$. **112**

5.22 Double shell model fits to the line integrated velocities in the toroidal plane for shot #7150 at $t = 59\mu s$.. ..... **113**

5.23 Jet diameter near the central midplane for shot #7150 at $t = 59\mu s$ calculated using the full width at quarter maximum of a gaussian fit to the intensity profile along the red line in the camera image. The resulting jet radius of 4.6 cm is larger than the initial guess for the double shell model but less than the average radius required by the model fits suggesting the possibility of a diffuse outer shell that is not apparent in camera images. **114**

5.24 Test of the matrix Abel inversion method using the double shell model with 20 km/s azimuthal velocity in the core, 10 km/s azimuthal velocity in the skin, and no axial flows. The model is presumed to be perfectly aligned with the $\hat{z}$ axis and bundle 7 is used for the inversion. a) Comparison of the inverted emissivity profile to the uniform emissivity profile of the double shell model. b) Comparison of the inverted azimuthal velocity profile to the double shell model. c) Graphical representation of the shells used in the inversion technique compared to the placement of the double shell model with a skin shell shown in orange and the core shell shown in red. Dashed lines represent the tangency radii of a given chord while red lines delineate the separate shells. **117**

5.25 Local azimuthal velocities for shot #7150 at $t = 59\mu s$ for bundles 7 and 8 using the matrix Abel inversion technique. The inversion was carried out for lines of sight above and below the center of the jet. The two free parameters, Δx and Δy, were adjusted from the initial guess location $(x, y, z) = (-0.047, -0.024, 0.53)$ $[m]$ obtained using gaussian fits to the binned intensity (Figure 5.18) until the magnitude of the velocity for the inner two view chords agreed. **118**

5.26 Local azimuthal velocities for shot #7150 at $t = 59\mu s$ for bundles 9-12 using the matrix Abel inversion technique. The inversion was carried out for lines of sight below the center of the jet only. The two free parameters, Δx and Δy, were adjusted from the initial guess location $(x, y, z) = (-0.047, -0.024, 0.53)\ [m]$ obtained using gaussian fits to the binned intensity (Figure 5.18) until the profiles were in good agreement. **119**

5.27 Linear fit to the collective toroidal velocity profiles obtained using a matrix Abel inversion technique for shot #7150 at $t = 59\mu s$. The results indicate a peaked azimuthal velocity profile with a shear of $\sim -6 \times 10^5 s^{-1}$ and azimuthal velocities on the order of ~ 65 km/s in the core of the jet. **122**

6.1 Response of cluster 2.16 of magnetic probe array 1 to a magnetic field applied in the $\hat{x}$ calibration direction at 300 kHz. Comparable pickup in the x and y chips indicates that they are not well aligned with the calibration axis. Calibration matrices constructed using this analysis correct for this misalignment. **126**

6.2 Frequency response of the main diagonals in the inverse calibration matrix for magnetic probe array 1. Calibration matrixes obtained using a 300kHz signal, corresponding to a period of $\sim 3\mu s$ were selected to be used for the analysis of shot data but can be swapped out easily to examine phenomena occurring at different timescales. Resonances at high frequencies were observed for several chips but was not systematic. **127**

6.3 Reconstructed magnetic fields for cluster 2.16 of magnetic probe array 1 using calibration matrix for 300kHz. Despite the weak calibration field, the reconstruction agrees with the applied field and corrects for the misaligned chips **128**

6.4 Magnetic field traces obtained from magnetic probe array 1 placed 21 cm in front of the LabJet gun for shot #6999. Filled regions indicate the associated calibration error. A bad chip for cluster 2.9 results in large errors and is therefore excluded from future analysis. **130**

6.5 Magnetic field evolution for shot #6999 using a single linear array placed 21cm in front of the gun electrodes shows a regular reversal in the magnetic field direction in all three orthogonal directions. a) High speed images taken with the Kirana camera at 1 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system. Alternating positive and negative axial magnetic fields (B_z) with comparable magnitudes (~ 0.4 T) suggest that the magnetic field structure of the long stable jet is not a simple screw pinch. **131**

6.6 Magnetic field evolution for a more collimated shot #7016 using a single linear array 21cm in front of the gun electrodes shows simultaneous strong positive and negative axial fields which flip about $x = 0$. a) High speed images taken with the Kirana camera at 1 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system..... **132**

6.7 Magnetic field profiles at various times for shot #7016 a) Magnetic field profile for the stage I jet at $t = 12$ μs. resembles a screw pinch. b) Magnetic field profile for the stage III jet with strong positive and negative axial fields not well described by a screw pinch. c) Negative helicity (i.e. $-\lambda$) Lundquist solution with a positive axial field. The profile is similar to the stage I jet but with the field components reversed. **133**

6.8 Magnetic field evolution for shot #7089 measured with the magnetic probe array in the central midplane also reveals a regular reversal of the magnetic field in each orthogonal direction. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system. **134**

6.9 Detached plasmoid propagates as a bow shock followed by a secondary shock associated with the propagation of the secondary jet a) High speed camera images show propagation of initial detached plasmoid and the start of jet formation. b) Magnetic probe data in each orthogonal direction as the bow shock propagates past the array. Four distinct regions are highlighted corresponding to a reversal in B_x and B_y. c) Diagram of the magnetic field reversal with the associated direction of the current sheet between each region. A perfect reversal in magnetic field between regions B and C is not always observed. **135**

6.10 Time of flight measurements based on the arrival of the first magnetic field signal shows that the initial plasma bow shock decelerates at $\sim 2 \times 10^6\ km/s^2$ as it propagates across the chamber. The formation and rapid deceleration of a bow shock suggest a deceleration mechanism such as field line tension of the vacuum field. **136**

6.11 Magnetic field evolution for shot #7175 measured with a single probe array at $z = 0.74$ m. In this shot, the jet remained well centered on the probe for $> 40\mu s$ before traversing upwards through the column. A regular reversal of the field directions and a gradual bending of the magnetic fields in the $-\hat{x}$ direction suggest an observational selection effect possibly related to rotation of the magnetic field structure. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system. **137**

6.12 Magnetic field evolution for shot #7176 measured with a single probe array at $z = 0.74$ m. In this shot the probe was observed to traverse downward through the plasma column and a large regular reversal of the magnetic field was again observed. In this case, the fields appear to bend gradually in the $+\hat{x}$ direction. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system.. ..... **138**

6.13 Time of flight measurements based on the arrival of the first magnetic field signal shows that the initial plasma bow shock decelerates at $\sim 2 \times 10^6\ km/s^2$ as it propagates across the chamber. The formation and rapid deceleration of a bow shock suggest a deceleration mechanism such as field line tension of the vacuum field.. ..... **139**

6.14 Method to reconstruct the cross section of a magnetic flux tube using a single magnetic probe array. The magnetic probe can traverse the tube in two directions (up-scan and down-scan) while measuring the magnetic field in all three orthogonal directions. One can therefore reconstruct the cross section of the flux tube (assuming axial uniformity during probe transit and alignment of the tube with the $\hat{z}$ direction) by overlaying contour plots of the axial field B_z with vector plots of the azimuthal field components B_x and B_y. To obtain the appropriate cross section, one must plot the ordinate with decreasing time values for down-scans, and increasing time values for up-scans. **140**

6.15 Cross section analysis for shot #6999 reveals opposing left handed flux tubes with large antiparallel currents and negative, non-uniform λ profiles. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y. b) Axial current and λ profiles overlaid on contours of axial magnetic field. c) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan of the probe array. d) Current density and λ profiles for flux tubes observed in the down-scan reconstruction **141**

6.16 Cross section analysis for shot #7089 with the probe array in the central midplane also reveals left handed flux tubes with large antiparallel currents and a negative λ. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan of the probe array. b) Current profile and λ profiles overlaid on contours of axial magnetic field. c) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for an upward scan of the probe array. d) Current and λ profiles for up-scan. The λ profile during this scan is relatively flat in both tubes with $\lambda \approx 140\ m^{-1}$..... **143**

6.17 Cross section analysis for shot #7175, which remained well centered on the probe array for $> 40\mu s$, suggests helical and/or rotating flux tubes. a) At $t \approx 65 - 70\mu s$ the magnetic probe array traverses upward through a positive axial flux tube and downward through a negative axial flux tube (black boxes). Later at $t \approx 85 - 90\mu s$ the probe array traverses downward through a positive axial flux tube and upward through a negative axial flux tube (red boxes). b) Interpretation of the results from the point of view of two helically intertwined flux tubes with opposite axial fluxes suggests that the helix rotates at ~ 25 kHz or equivalently the magnetic probe array translates axially through the helix resulting in an apparent rotation in the magnetic probe data (or both). **144**

6.18 Helical $m = 1$ Taylor state model used to compare with experimental data. The model has an aspect ratio of 25 with $\lambda a = 3.119$. The model does not rotate or oscillate in time such that the hypothetical probe trajectory is solely responsible for changes in the observed field. **146**

6.19 Comparison of experimental magnetic field data from shot #7150 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model. **147**

6.20 Comparison of experimental magnetic field data from shot #7153 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for an upward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model. **149**

6.21 Comparison of experimental magnetic field data from shot #7175 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y presuming an upward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model..... **150**

7.1 Comprehensive picture of the long, stable, collimated jets formed in the MoCHI.LabJet experiment based on experimental observations presented in this thesis. The core of the jet consists of an isolated, high-β, double-helix magnetic field with negative helicity content which rotates in the $+\hat{\phi}$ direction like an Archimedes screw. This structure is bordered by a hotter sheath plasma (orange) that rotates in the azimuthal direction with a lower velocity than the core. Strong positive axial flows are also observed and are presumed to flow within both volumes. The helical shear flow is sufficient to stabilize the current-carrying magnetized jet to kink instabilities..... **157**

7.2 Astrophysical jets with a double helix structure. a) The Double Helix Nebula near the center of our own Milky Way Galaxy. b) Quasar 3C273 with a $\sim$200,000 light year long helical jet boxed in red. Image credits: Nasa/JPL – Caltech/M. Morris (UCLA), ESA/Hubble.. ..... **158**

A.1 Fast ionization gauge cathode-anode circuit (above). Simplified differentiator op-amp circuit for ion current measurements (below). **169**

A.2 Solid works diagram of the fast ionization gauge vacuum design (above). Corresponding picture of the completed probe (below). **171**

A.3 Calibration curve of V_{out}/I_e as a function of chamber pressure for three calibration tests. Curves were not linear as expected, and were best fit using a power law curve that minimized the total percent error such that the low-pressure values were weighted equally. **172**

A.4 Reconstructed FIG pressures for each of the three calibration runs. The Reconstructed pressures are in good agreement with the actual chamber pressure over a range of $\sim 5 \times 10^{-7}$ - 3×10^{-4} Torr. **173**

A.5 Characterization of the Parker Series 9 valve driven by a custom fast valve drive unit (FVDU). Different pulse lengths resulted in differing throughputs and opening times, however all proved to be insufficient..... **174**

A.6 Characterization of the 10 custom fast gas valves at their respective operational positions compared to the Bellan valves after which they were designed. For this test the FIG detector head was approximately 15 cm in front of the center of the LabJet gun. **175**

A.7 Evolution of a gas puff in front of the LabJet gun electrodes with a single valve connected to the middle and outer-inner gas injection plenums with a back-line pressure of 100 PSI and gas valve PSU voltage of 1kV. The red lines at the bottom of the plot represent the electrodes with gas slits and high voltage insulating gap shown **176**

B.1 Custom 3D printed 32-channel V-groove clamp. Bottom of the clamp (left) and top of clamp (right) are virtually identical except for hexagonal holes to house the clamping nuts on the bottom clamp. A larger cavity provides some room for excess fiber between the surethane tubing barb (far side) and the v-groove chips (near side)..... **179**

B.2 Custom 3D printed 8 channel V-groove clamp; bottom (left) and top (right). Protective tubing is secured via a ¼" tubing barb in the center of the clamp and the v-groove is secured between the threaded portion of the clamp. Collimating lenses mounted in ¼" gas line compression nuts can be screwed directly onto the clamp to adjust the focus of the fibers. **179**

B.3 12-way fiber bundle breakout clamp; bottom (left) and top (right). Polyethylene tubing for all 12 breakout bundles is secured with ¼" tubing barbs on the far side of the breakout clamp while the ¾" surethane tubing housing all fibers is secured with a custom 3D printed barb on the near side of the clamp (see figure B.4). **180**

B.4 Custom 3D printed barb used to secure the ¾" surethane tubing in the 12-way fiber bundle breakout clamp and 32 channel v-groove clamp. **180**

B.5 Ray tracing diagram for plasma side collimating optics. The acrylic bi-aspheric lens has a focal length of 4.6mm. In this diagram, the fibers are mounted 0.03mm behind the focal point resulting in a magnification of -153. The depth of field extends from the vacuum chamber port entrance to infinity. **181**

B.6 Photographs taken during the manufacture of the custom fiber bundle. See text for a full description, the numbers of the highlighted photographs correspond the manufacturing steps listed in the text. **182**

LIST OF TABLES

Table Number **Page**

2.1 Definitions of canonical quantities in each regime **21**

2.2 Field quantity definitions **22**

3.1 Poloidal flux functions and flux coordinates for pertinent vector fields **29**

3.2 Selected jet parameters for kinematic evolution calculations and corresponding figures .. **38**

4.1 Comparison of experimental and astrophysical jet parameters **59**

5.1 Angle between view chords in the poloidal plane and the $\hat{z}$ axis **93**

5.2 Model fit parameters for shot #7150 in the poloidal plane at $t = 43\mu s$ **110**

5.3 Model fit parameters for shot #7150 in the toroidal plane at $t = 43\mu s$..... **111**

5.4 Model fit parameters for shot #7150 in the poloidal plane at $t = 59\mu s$ **112**

5.5 Model fit parameters for shot #7150 in the toroidal plane at $t = 59\mu s$..... **113**

Chapter 1
INTRODUCTION

1.1 The Plasma Universe

As humans, we occupy a unique region of space where solids, liquids, and gases comprise the dominant states of matter. These states can generally be described according to how the constituent atoms interact. In a solid, interatomic forces are strong and atoms are constrained to vibrate within a rigid crystal lattice. When a solid is heated sufficiently, the thermal motion of the atoms breaks the crystal lattice structure and (usually) a liquid is formed. When a liquid is heated such that atoms vaporize off the surface faster than they recondense, a gas is formed. If the heating continues however, atoms within a gas can collide with sufficient energy that electrons are stripped away and the atoms become ionized. In its simplest description, this collection of charged and neutral particles constitutes the fourth state of matter known as plasma. While the transition between a weakly ionized gas and a plasma is largely a matter of terminology, a more rigorous definition of plasma requires quasineutrality (i.e. zero net electric charge over scale lengths larger than the Debye screening length λ_D) and collective behavior (i.e. plasma frequency is large compared to electron-neutral collision frequency) [1]. It is this collective behavior that differentiates plasmas from the other states of matter and gives rise to phenomena of incredible complexity, utility, and oftentimes great beauty.

Despite its relative rarity on earth, it is estimated that 99% of the visible matter in the universe exists in a plasma state [2]. As such, plasma physics provides the basis for the dynamics of astrophysical systems from the very largest relativistic jets to planetary auroras. Within earth's atmosphere, familiar examples of plasmas include lightning, fluorescent lights, and arc discharges. More lofty applications include spacecraft propulsion, laboratory astrophysics, and fusion energy research. Not all plasmas are created equal however, and generally they are characterized according to three fundamental parameters: particle density, measured in particles per unit volume; species temperature (i.e. ion and electron temperatures), typically measured in electron-Volts (eV); and magnetic field strength, measured in Tesla (T). To ensure quasineutrality, the characteristic dimension of a plasma must be larger than the Debye screening length, the distance over which the influence of the electric field of an individual particle or charged surface is felt by other charged particles inside the plasma. Figure 1.1 demonstrates the vast range in scale-length, density, and

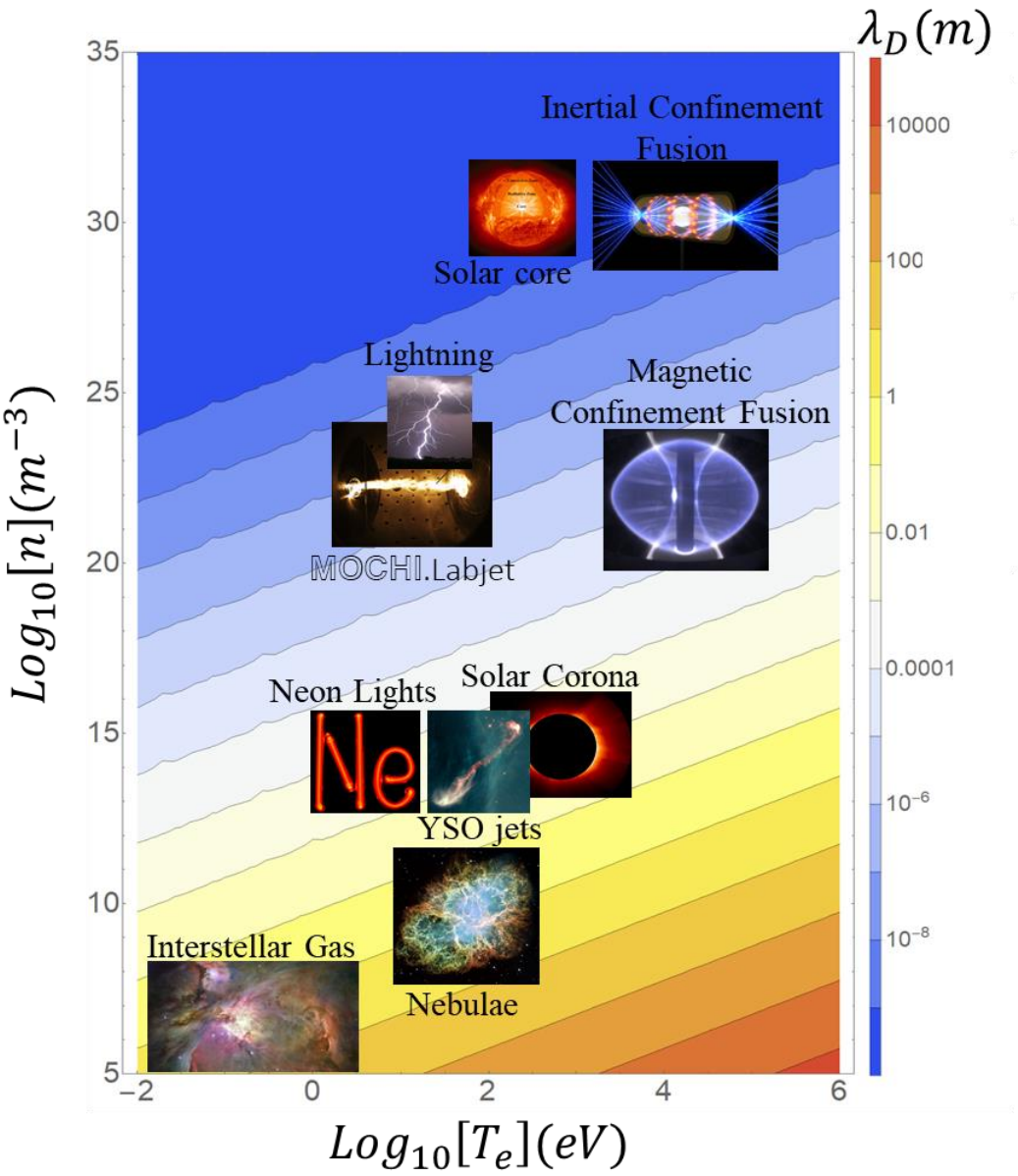


Figure 1.1: Various naturally occurring and laboratory plasmas and their position in density-temperature space. For an ionized gas to be considered a plasma, one key requirement is that the characteristic dimensions must be much larger than the Debye length $\lambda_D = \sqrt{(\epsilon_0 k_B T_e / n_0 q_e^2)}$ which is the distance at which charge is shielded within a plasma. For most terrestrial plasmas, the Debye length $< 1mm$. For astrophysical plasmas however, the Debye length can be tens of meters. Image Credits: Interstellar Gas, NASA/Hubble; Nebulae, NASA/ESA/J. Hester and A. Loll (Arizona State University); Solar Corona, ESA & NASA/SOHO; Lightning, NOAA; Magnetic Confinement Fusion, MAST tokamak, Culham centre for fusion energy; Solar Core, SOHO/EIT consortium; Inertial confinement Fusion, Lawrence Livermore National Laboratory. YSO jets, J. Morse/STSci, and NASA/ESA.

temperature of various plasmas as well as the associated Debye lengths. This broad span of parameters makes it nearly impossible to apply a single theory towards a unified description of plasma behavior. In fact, even within a single plasma, phenomena at disparate spatial and temporal scales often require different models to accurately and efficiently predict and understand. For this reason, plasma physics is rarely a precise science, but rather a jumble of overlapping viewpoints, each accounting for a limited range of behaviors. A full understanding is therefore developed by studying these various points of view while keeping in mind the various connections between these perspectives.

1.2 The many descriptions of plasmas

Because plasmas consist of charged particles, in principal, plasma behavior can be determined by the self-consistent interaction of these particles with the electromagnetic fields they inhabit and create. This relationship is illustrated schematically in figure 1.2 where the instantaneous force on each particle j can be evaluated using the Lorentz equation and the resulting electric and magnetic fields ($\vec{E}$ and $\vec{B}$ respectively) can be determined from Maxwell's equations based on the trajectory of all the constituent particles. Because each particle is considered independently, plasmas that require this level of description to accurately describe their dynamics are considered to occupy the single particle regime. While this framework is conceptually easy to understand, the sheer number of particles involved in most plasmas and the wide range of spatial and temporal scales at which the particles interact makes such calculations virtually intractable. Consequently, the next several sections will outline the three primary models used to describe plasmas, organized from most to least detailed.

1.2.1 A kinetic description of plasmas

Following the single particle description of plasma dynamics, a subsequent model arises as an extension to the kinetic theory of gases. To understand this model, one must introduce a concept known as phase-space. The phase space of a three-dimensional mechanical system consists of a six-dimensional volume where each point within the volume corresponds to a unique value of position and velocity (or momentum). At any given time in a plasma, every particle occupies a specific position in phase space. One can therefore characterize the instantaneous configuration of many particles by specifying the density of particles at each point in phase-space, this quantity is

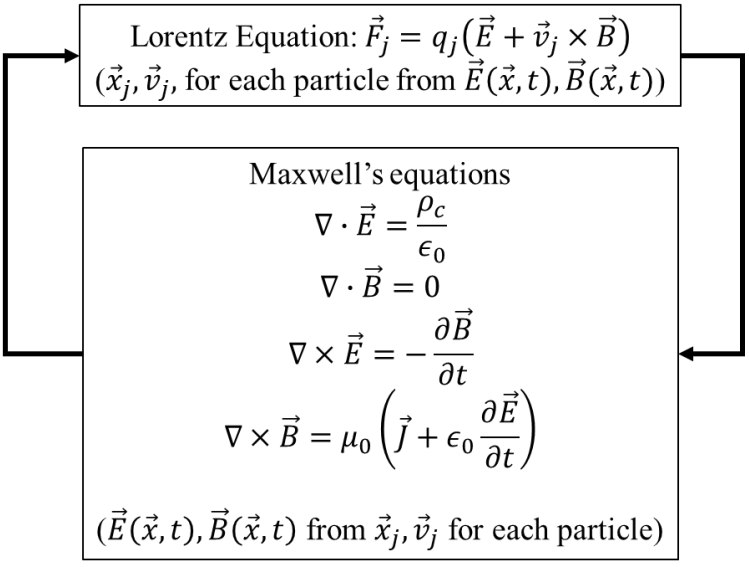


Figure 1.2: Relationship between individual particle trajectories and the electromagnetic fields. If all fields and particle trajectories are known, the evolution of the plasma can be determined exactly.

known as the distribution function $f_\sigma(\vec{x}, \vec{v}, t)$ where $\sigma = e, i$ corresponds to electron and ion species respectively. The motion of the ensemble of phase space points can be described using the Boltzmann equation [1,3]

$$\frac{\partial f_\sigma}{\partial t} + \vec{v}_\sigma \cdot \nabla_x f_\sigma + \frac{q_\sigma}{m_\sigma}\left(\vec{E} + \vec{v}_\sigma \times \vec{B}\right) \cdot \nabla_v f_\sigma = \left.\frac{\partial f_\sigma}{\partial t}\right|_{collisions} \tag{1.1}$$

which, in conjunction with Maxwell's equations, provides a complete description of plasma evolution. However, solving these kinetic equations in seven dimensions with details of single particle motion entering in the collision terms is again a formidable, and often intractable task for most plasmas. For this reason, it is often necessary to replace the detailed velocity distribution with only the most elementary information: the mean velocity (i.e. drift velocity) and variance (i.e. temperature). This information can be completely described using a Maxwellian distribution

$$f_\sigma(\vec{v}) = n_0\left(\frac{m_\sigma}{2\pi T}\right)^{\frac{3}{2}} e^{\left(-\frac{m_\sigma(\vec{v}-\vec{v}_\sigma)^2}{2T}\right)} \tag{1.2}$$

which solves Boltzmann's equation when $\partial f_\sigma / \partial t$ due to collisions is zero. Because the cumulative effect of collisions is randomizing, this assumption therefore suggests that each species has already achieved thermodynamic equilibrium. This assumption is required as we move towards a

macroscopic description of plasmas yet precludes kinetic effects associated with non-Maxwellian velocity distributions.

1.2.2 A multifluid description of plasmas

Because the distribution function $f_\sigma(\vec{x}, \vec{v}, t)$ describes the density of particles in phase space, it is easy to observe that if one were to integrate this function over all velocities at a given time the resulting function gives the number of particles at a given position $n_\sigma(\vec{x}, t)$. Furthermore, if one considers the distribution function as the probability that a randomly selected particle at position $\vec{x}$ and time t has a velocity $\vec{v}$, then averaging over the velocities of all particles at a given position gives the mean velocity $\vec{u}(\vec{x}, t)$ where $n(\vec{x}, t)\vec{u}(\vec{x}, t) = \int \vec{v} f(\vec{x}, \vec{v}, t) d\vec{v}$. Continuing this procedure, if we multiply $f(\vec{x}, \vec{v}, t)$ by $\vec{v}^2$ and integrate over velocity space, we arrive at an expression for the mean energy of the particles as a function of position and time. This process of multiplying the distribution function by powers of $\vec{v}$ and integrating over velocity space is called taking the moments of the distribution function. It is this procedure that is used to derive macroscopic quantities from a microscopic description of particles in phase space.

To derive a fluid description of plasmas, one proceeds by taking moments of the entire Boltzmann equation. Each moment introduces a new unknown whose temporal evolution is described by the next moment of the Boltzmann equation. In theory, this infinite chain of moment equations completely describes a plasma without any assumptions placed on the velocity distribution function. To be useful in practice however, this chain is truncated, typically after the first 2 moments and therefore only captures the evolution of moments associated with the Maxwellian velocity distribution. Higher order kinetic effects can be captured by taking higher moments of the Boltzmann equation, however these models will not be discussed here.

The resulting fluid equations from moments 0 through 2 of the Boltzmann equation are, respectively:

Continuity:
$$\frac{\partial n_\sigma}{\partial t} + \nabla \cdot (n_\sigma \vec{u}_\sigma) = 0 \tag{1.3}$$

Momentum:
$$\rho_\sigma \frac{D\vec{v}_\sigma}{Dt} + \nabla P_\sigma + \nabla \cdot \overleftrightarrow{\Pi}_\sigma - q_\sigma n_\sigma \left(\vec{E} + \vec{v}_\sigma \times \vec{B}\right) = \sum_{\substack{\alpha \\ \alpha \neq \sigma}} \vec{R}_{\sigma\alpha} \tag{1.4}$$

Energy: $$\frac{3}{2}n_\sigma \frac{DT_\sigma}{Dt} + P_\sigma \nabla \cdot \vec{v}_\sigma + \overleftrightarrow{\Pi}_\sigma : \nabla \vec{v}_\sigma + \nabla \cdot \vec{h}_\sigma = \sum_{\substack{\alpha \\ \alpha \neq \sigma}} Q_{\sigma\alpha} \quad (1.5)$$

where $\rho_\sigma = n_\sigma m_\sigma$ is the species density, P_σ is the isotropic scalar pressure, $\overleftrightarrow{\Pi}_\sigma$ is the anisotropic stress, $\vec{R}_{\sigma\alpha}$ is the momentum transfer between species due to collisions, $\vec{h}_\sigma$ is the heat flux, and $Q_{\sigma\alpha}$ is the heat generated by random collisions of unlike particles. Maxwell's equations are then used to couple the electromagnetic field to the plasma fluids. To close the system, some sort of ad hoc closure procedure must be invoked. Typical closure schemes involve invoking adiabatic ($P_\sigma \propto n_\sigma^\gamma$) or isothermal assumptions ($T_\sigma = constant, P_\sigma \propto n_\sigma$).

For highly ionized plasmas, it is often assumed that there are only two plasma species, ions and electrons ($\sigma = i, e$). Equations 1.3 – 1.5 when evaluated for each species therefore establish the basis of what is known as the two-fluid model. Because the only assumption made to this point is a Maxwellian velocity distribution function, this model contains all the physics necessary to describe plasma behavior except for kinetic effects associated with a Non-Maxwellian distributions.

1.2.3 A single-fluid description of plasmas

In the multifluid system, particle motion is described by individual species mean velocities and their pressures. However, an alternate single-fluid description can be constructed by introducing two new velocity variables that are a linear combination of the species velocities. These velocity variables are the current density $\vec{j} = \sum_\sigma n_\sigma q_\sigma \vec{u}_\sigma$ and the center of mass velocity $\vec{U} = \frac{1}{\rho}\sum_\sigma m_\sigma n_\sigma \vec{u}_\sigma$ where $\rho = \sum_\sigma m_\sigma n_\sigma$ is the total mass density. Using these definitions, we can derive a set of single fluid magnetohydrodynamic (MHD) equations by taking moments of Boltzmann's equation, multiplying by m_σ and summing over species. The results of this procedure are [3]:

Continuity: $$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{U}\right) = 0 \quad (1.6)$$

Momentum: $$\frac{\partial \left(\rho \vec{U}\right)}{\partial t} + \nabla \cdot \left(\rho \vec{U}\vec{U}\right) = \left(\sum_\sigma n_\sigma q_\sigma\right)\vec{E} + \vec{J} \times \vec{B} - \nabla \cdot \overleftrightarrow{P}_{MHD} \quad (1.7)$$

where $\overleftrightarrow{P}_{MHD} = \overleftrightarrow{P}_e + \overleftrightarrow{P}_i$. However, because we are interested in producing a simpler description of plasma behavior, several simplifying assumptions are typically invoked. One such simplifying assumption is that the plasma behaviors of interest take place at a sufficiently large spatial scale and time scale that charge separation is small and electrons respond quickly to maintain quasineutrality. ($\sum_\sigma n_\sigma q_\sigma = 0$). With this approximation, equation 1.7 can be rewritten as:

$$\rho \frac{D\vec{U}}{Dt} = \vec{J} \times \vec{B} - \nabla \cdot \overleftrightarrow{P}_{MHD} \tag{1.8}$$

providing one equation relating $\vec{J}$ and $\vec{U}$. To provide a second equation, the electron momentum equation (1.4 with $\sigma \to e$) is considered in the low frequency, large spatial scale limit. If the characteristic time scales of interest are long compared to the electron cyclotron frequency, then the electron inertia term $m_e \frac{D\vec{u}_e}{Dt}$ can be dropped and applying the expression $\vec{R}_{ei} = en_e\eta\vec{J}$ (where η is the electrical resistivity) equation 1.4 reduces to

Gen. Ohm's Law: $$\vec{E} + \vec{U} \times \vec{B} = \eta\vec{J} + \frac{1}{Zen}(\vec{J} \times \vec{B} - \nabla P_e - \nabla \cdot \overleftrightarrow{\Pi}_e) \tag{1.9}$$

where the $\vec{J} \times \vec{B}$ term is called the hall term and is often neglected when the pressure term in equation 1.8 is negligible compared to the other two terms, or when the electron-ion collision frequency is large compared to the electron cyclotron frequency. The thermal term due to electron pressure gradients is also frequently ignored, resulting in a simplified Ohm's law for a conducting single fluid.

Ohm's Law: $$\vec{E} + \vec{U} \times \vec{B} = \eta\vec{J} \tag{1.10}$$

The magnetohydrodynamic energy equation can be derived by adding the electron and ion energy equations and manipulating to yield:

Energy: $$\frac{d}{dt}\left(\frac{P}{\rho^\gamma}\right) = \frac{\gamma - 1}{\rho^\gamma}\left[Q_{ie} + Q_{ei} - \nabla \cdot \left(\vec{h}_i + \vec{h}_e\right) - \overleftrightarrow{\Pi}_i : \nabla\vec{v}_i - \overleftrightarrow{\Pi}_e : \nabla\vec{v}_e\right] + \frac{\vec{J}}{Zen} \cdot \nabla\left(\frac{P_e}{\rho^\gamma}\right) \tag{1.11}$$

which must be closed with an equation of state that typically invokes adiabatic, isothermal, or cold plasma assumptions. In the case of adiabatic assumptions, the left side of equation 1.10 goes to zero and the energy equation becomes $\frac{P}{\rho^{\gamma}} = constant$. In conjunction with Faraday and Ampere's law, equations 1.6, 1.8, 1.10, and 1.11 provide the basis for a single-fluid magnetohydrodynamic description of plasmas. Because of the simplifying assumptions that were made, this description is far more tractable than multi-fluid, kinetic, or single particle descriptions and for that reason it is the most popular model for describing plasma behavior (provided the dynamics of interests satisfy the asymptotic approximations). The next section will therefore outline some of the important consequences that arise from this single fluid description.

1.3 Magnetohydrodynamic (MHD) theory

1.3.1 Frozen-in Flux Condition

For the case of an ideal, infinitely conducting plasma, the right-hand side of equation 1.10 is zero and we are left with the ideal form of Ohms law. One important consequence that arises in such a scenario is that the magnetic flux $\psi = \int \vec{B} \cdot d\vec{S}$ becomes frozen-in to the frame of the moving plasma. This can be demonstrated by looking at the time rate of change of magnetic flux through a surface

$$\left.\frac{d\psi}{dt}\right|_{S} = \iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} - \oint \vec{v} \times \vec{B} \cdot d\vec{l} \tag{1.12}$$

which consists of a term due to time changing magnetic fields as well as a Leibniz term due to motion of the surface with a velocity $\vec{v}$. By taking the curl of ideal Ohm's law, and using faradays law we can express the surface integral as a contour integral by applying stokes theorem

$$\begin{aligned} \iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} &= \iint \nabla \times \left(\vec{U} \times \vec{B}\right) \cdot d\vec{S} \\ &= \oint \left(\vec{U} \times \vec{B}\right) \cdot d\vec{l} \end{aligned} \tag{1.13}$$

where the contour is the boundary of the surface *S*. Plugging 1.13 into 1.12 gives

$$\frac{d\psi}{dt} = \oint \left(\vec{U} - \vec{v}\right) \times \vec{B} \cdot d\vec{l} \tag{1.14}$$

which is equal to zero when $\vec{U} = \vec{v}$ (i.e. when the surface moves with the plasma). Because the rate of change of flux is zero, we conclude that the magnetic flux is "frozen-in" to the plasma, and they move together. This is a tremendously important result, as it implies that to track the global evolution of the plasma, it is sufficient to track the shape and evolution of tubes of constant magnetic flux (henceforth referred to as magnetic flux tubes). Non-ideal effects such as Hall terms, pressure tensors, finite Larmor radius effects, kinetics, and particle inertia are generally assumed to be localized to distinct control volumes and treated separately as the need arises due to the increased computational cost and complexity.

1.3.2 Magnetic Helicity

As geometric objects, magnetic flux tubes have distinct topological properties (i.e. twists, writhes, links) that can be quantified within a given volume using a quantity known as magnetic helicity [4]

$$\mathcal{K} \equiv \iiint \vec{A} \cdot \vec{B}\, dV \tag{1.15}$$

where $\vec{A}$ is the magnetic vector potential and $\vec{B} = \nabla \times \vec{A}$ is the magnetic field. For an ideal, infinitely conducting plasma in an isolated volume with no open field lines it is straightforward to demonstrate that this quantity is conserved if the gauge of $\vec{A}$ is suitably chosen such that the scalar potential vanishes [4]. Although subtle, the issue of gauge dependence is crucial to the definition of $\mathcal{K}$, where the magnetic helicity can only be gauge independent (and therefore have a well-defined physical meaning) if no magnetic flux pierces the volume [5]. This ambiguity in gauge corresponds

to ambiguity in counting the amount of twists, writhes, and links of magnetic flux tubes external to the integration volume (Fig. 1.3).

To account for gauge ambiguity and provide a well-defined, physically meaningful description of helicity within a volume with open field lines, one can define a gauge-invariant relative magnetic helicity which depends only on quantities defined inside the bounding volume [5, 6]. This relative magnetic helicity can be expressed as

$$\mathcal{K}_{rel} \equiv \iiint \vec{A}_+ \cdot \vec{B}_-\, dV \tag{1.16}$$

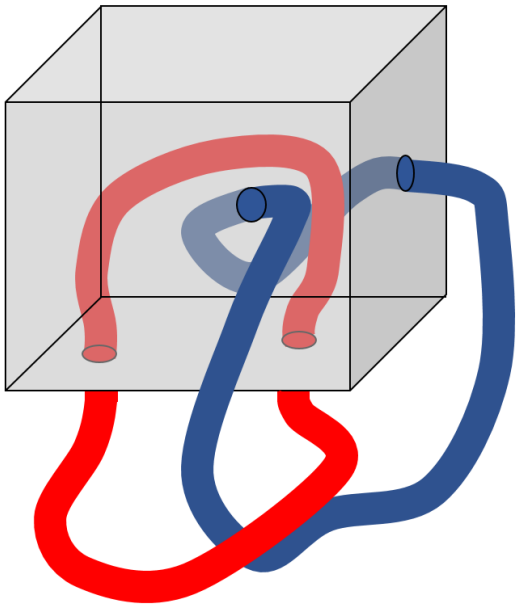


Figure 1.3: Representation of gauge ambiguity for two intertwined magnetic flux tubes within a bounding volume V (grey box). If magnetic flux leaves the bounding volume, the magnetic helicity integral cannot count the number of linkages, writhes and twists of the two flux tubes external to the volume. Adapted from [5].

where the $+/-$ subscripts represent the addition and subtraction of reference fields ($i.e. \vec{A}_{+} = \vec{A} + \vec{A}_{ref}$) that satisfy the following conditions

$$\begin{cases} \nabla \times \vec{B}_{ref}\big|_V = 0 \\ \vec{B} \cdot \hat{n}|_S = \vec{B}_{ref} \cdot \hat{n}|_S \\ \nabla \times \vec{A}_{ref} = \vec{B}_{ref} \end{cases} \tag{1.17}$$

where $\hat{n}$ is the unit normal vector of the bounding volume. In a flux conserving volume, the rate of change of relative magnetic helicity is

$$\frac{d\mathcal{K}_{rel}}{dt} = -2 \iiint \vec{E} \cdot \vec{B} dV - 2 \oiint \phi \vec{B} \cdot d\vec{S} \tag{1.18}$$

where ϕ is the electrostatic potential. The first term on the right corresponds to a helicity dissipation term and the second term corresponds to a helicity flux into the volume due to an applied electrostatic potential on the boundary. For the case of an ideal, infinitely conducting

plasma ($\vec{E} = 0$) within an equipotential flux conserving volume ($\phi = const$ on S), the total relative magnetic helicity is conserved.

1.3.3 Constrained relaxation and force-free states

The importance of magnetic helicity becomes apparent when we consider its ruggedness in the presence of dissipative effects; where ruggedness is the invariance of a quantity with respect to the energy of the system. For a magnetically dominated MHD plasma (i.e. low β plasma, where β is the ratio of thermal pressure to magnetic pressure), the free energy is almost entirely magnetic

$$W = \iiint \frac{B^2}{2\mu_0} dV \tag{1.19}$$

and is perfectly conserved for an ideal, infinitely conducting plasma in an isolated volume. In the presence of dissipation however, magnetic flux is no longer perfectly frozen-in and some dissipation of magnetic energy is required. By comparing the dimensions of magnetic energy to magnetic helicity, it can be shown that

$$\frac{\delta W}{\delta \mathcal{K}} \propto \frac{1}{L} \tag{1.20}$$

where L is the characteristic linear dimension of the dissipative phenomena. Therefore, dissipative phenomena occurring at small scales will preferentially dissipate magnetic energy over magnetic helicity. One such phenomena that is intrinsically small scale, is magnetic reconnection, where magnetic field lines break and reconnect to form a new topology. In the limit that the reconnection scale becomes infinitesimally small, helicity dissipation becomes negligible compared to magnetic energy. The magnetic energy cannot decay to zero however, if the helicity is conserved, and in 1959 Woltjer [4] was the first to describe this constrained dissipation in the form of a variational problem

$$\delta W - \lambda \delta \mathcal{K} = 0 \tag{1.21}$$

where λ plays the role of a Lagrange multiplier. If we assume that the magnetic field is of the form $\vec{B} + \delta\vec{B}$, where $\delta\vec{B} = \nabla \times \delta\vec{A}$ and $\vec{B}$ is the magnetic field that minimizes W equation 1.21 becomes

$$\begin{aligned} &\iiint \frac{\left(\vec{B}\cdot\overrightarrow{\delta B}\right)}{\mu_0}dV \;-\lambda\iiint\left(\vec{A}\cdot\left(\nabla\times\overrightarrow{\delta A}\right)+\overrightarrow{\delta A}\cdot\vec{B}\right)dV=0 \\ \Rightarrow\;&\oiint\left(\overrightarrow{\delta A}\times\vec{B}\right)\cdot d\vec{S}+\iiint\overrightarrow{\delta A}\cdot\left(\nabla\times\vec{B}\right)dV-\lambda\oiint\left(\overrightarrow{\delta A}\times\vec{A}\right)\cdot d\vec{S} \\ &\qquad-2\lambda\iiint\overrightarrow{\delta A}\cdot\vec{B}dV=0 \end{aligned} \tag{1.22}$$

where we have absorbed the factor of μ_0 into λ in the second line. For the case of a conducting, flux conserving boundary ($\vec{\mathrm{B}}\cdot d\vec{S}=0,\ \overrightarrow{\delta A}_{\parallel}=0$) the surface integrals vanish. Since the relation must hold for any arbitrary $\overrightarrow{\delta A}$ within the volume we are left with the condition

$$\nabla\times\vec{\mathrm{B}}=\lambda\vec{B} \tag{1.23}$$

where the current density $\vec{J}=1/\mu_0(\nabla\times\vec{B})$ is always parallel or antiparallel to the magnetic field. States that satisfy equation 1.23 are known as constant λ "force-free" states, since the $\vec{J}\times\vec{B}$ force in equation 1.8 is zero (and $\nabla\cdot\overleftrightarrow{P}_{MHD}=0$ was assumed with the low β approximation). Although λ does not have to be a constant within the volume for a plasma to be considered force-free, the above analysis demonstrates that constant λ states are of particular significance as they represent the lowest magnetic energy state which a closed, ideal MHD system can attain and are therefore the natural end configuration of a system with dissipation [7].

Taking the curl of equation 1.23 yields a vector Helmholtz equation

$$\nabla^2\vec{B}+\lambda^2\vec{B}=0 \tag{1.24}$$

which shows that λ is also an eigenvalue of the system related to the characteristic dimensions of the bounding volume. To satisfy equation 1.24 with a prescribed set of boundary conditions, only certain discrete values of λ are allowed. Furthermore, since $\lambda\propto W/\mathcal{K}$, the lowest energy state of a system corresponds to a state with the lowest allowed λ. Performing a similar minimization for an open or driven configuration using $\delta\mathcal{K}_{rel}$ as opposed to $\delta\mathcal{K}$ in equation 1.21 also results in equation 1.23; however, because of the change in surface boundary conditions λ is no longer a dependent parameter.

The simplest solution to equation 1.23 was discovered by Lundquist [8] using a cylindrical coordinate system with the assumption of azimuthal symmetry. The solution

$$\begin{aligned} B_r &= 0 \\ B_\phi &= B_0 J_1(\lambda r) \\ B_z &= B_0 J_0(\lambda r) \end{aligned} \tag{1.25}$$

describes helical field lines on an infinite cylinder. Since the Bessel function is oscillatory in r, there are both left and right-handed helices with positive and negative B_z. By restricting consideration to a volume such that $B_z = 0$ at the surface $r = a$, one obtains the field line topology shown in figure 1.4.

Considerations subsequent to Lundquist's soon led to general solutions for force-free magnetic fields satisfying equation 1.23 [9,10,11,12]. One such representation in cylindrical coordinates takes the form [12]

$$\vec{B} = \hat{z} \times \nabla\psi + \frac{1}{\lambda}\nabla \times (\hat{z} \times \nabla\psi) \tag{1.26}$$

where ψ satisfies the scalar Helmholtz equation $(\nabla^2 + \lambda^2)\psi = 0$. Using separation of variables, the general solution that satisfies the boundary conditions that $B_z = 0$ at $z = 0, L$ are of the form

$$\psi = \left(\sum_{n=1}^{\infty} A_n J_m(k'a) \sin(kz) + A_0 r^m e^{i\lambda z} \right) e^{im\theta} \tag{1.27}$$

where $k' = \sqrt{\lambda^2 - k^2}$ and $k = n\pi/L$. Applying the boundary condition $B_r(a) = 0$ results in the following condition

$$\sum_{n=1}^{\infty} A_n \left(\frac{im}{a} J_m(k'a) \sin(kz) + \frac{kk'}{\lambda} J'_m(k'a) \cos(kz) \right) + 2imA_0 a^{m-1} e^{i\lambda z} = 0 \tag{1.28}$$

for all z between 0 and L. For azimuthally symmetric modes ($m = 0$), only one coefficient A_n is non-zero and we have

$$\begin{aligned} B_r &= kJ_1(k'r)\cos(kz) \\ B_\phi &= -\lambda J_1(k'r)\sin(kz) \\ B_z &= -k'J_0(k'r)\sin(kz) \end{aligned} \tag{1.29}$$

where $k'a = j_{l1}$, the lth zero of J_1, and $\lambda = \pm\sqrt{k^2 + j_{l1}^2/a}$. The minimum energy state (corresponding to the lowest value of λ) is for $n = l = 1$, and is known as the cylindrical

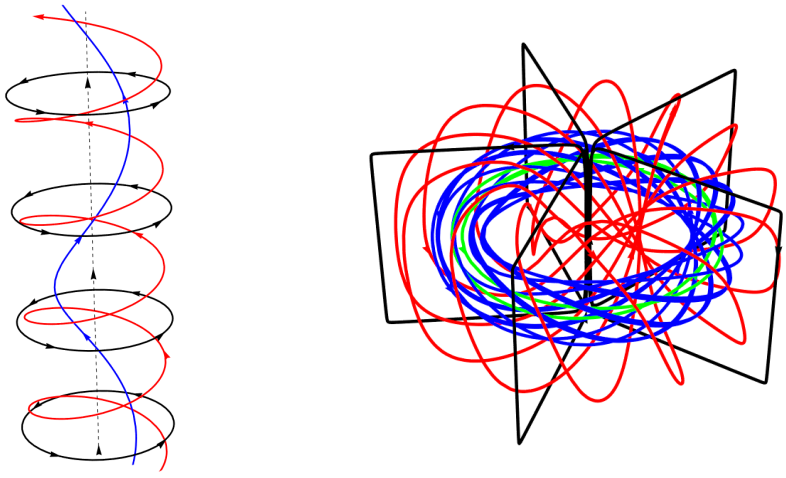


Figure 1.4: Magnetic field line topology of the Lundquist solution in an infinite cylinder (left) and cylindrical spheromak in a flux conserver with an aspect ratio $L/a = 1$ (right). The spheromak fields form toroidal helices around closed magnetic flux surfaces and are purely poloidal ($\hat{r}, \hat{z}$) at the boundaries (black lines).

spheromak (Fig. 1.4). Similar analysis in spherical coordinates [11] also yield spheromak solutions. For doubly connected volumes (i.e. donut like vessels), solutions resemble Lundquist's solution wrapped around on itself and are known as reverse-field pinches (RFP) due to the spontaneous reversal of the azimuthal field near the wall.

For non-axisymmetric ($m \neq 0$) solutions, the radial boundary condition (equation 1.28) cannot be satisfied by a single n because J_m and J'_m cannot both be zero for the same argument. In fact, this condition cannot be satisfied by any finite number n since any such finite set of trigonometric functions are not linearly independent. To construct an approximate solution, one can truncate equation 1.28 at $n = N$, giving an equation with $N + 1$ complex unknowns (since $A_0, A_1, \ldots A_n$ are complex) [13]. Evaluating at $N + 1$ locations on the interval $0 < z < L$ gives a set of equations which can be cast in matrix form where the determinant of the resulting $(N + 1) \times (N + 1)$ complex matrix is a complicated function of λ. The condition that the determinant be zero specifies allowed values of λ for the given geometry. Because the summation was truncated at finite N, the determinant will never be exactly zero; however, it will display pronounced minima at the valid λ solutions as well as spurious roots appearing at $\lambda = n\pi/L$ (Figure 1.5a). To accurately predict the correct λ values it is necessary to truncate the summation

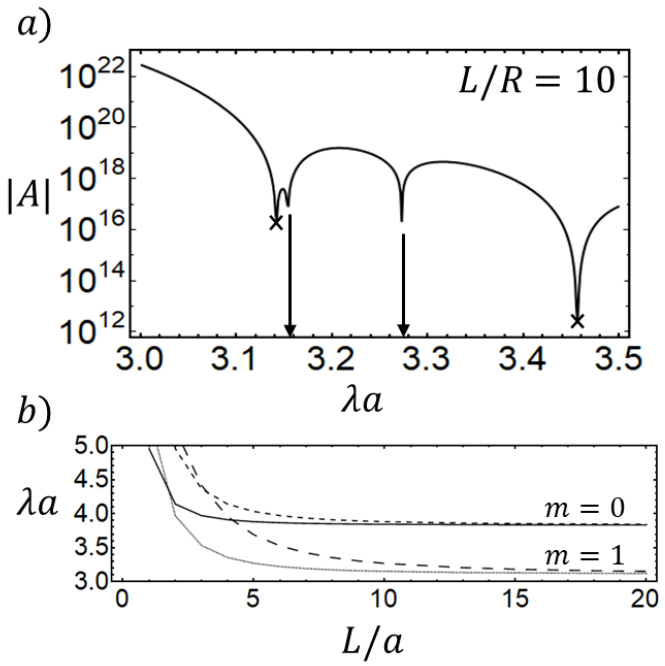


Figure 1.5: (a) The determinant of equation 1.28 evaluated for an aspect ratio of 10 and $m = 1$. The first two physical roots appear at $\lambda a = 3.15$ and $\lambda a = 3.27$. (b) Dependence of λa on the aspect ratio of the flux conserving volume for the first two solutions with $m = 0$ and $m = 1$ symmetry. The minimum energy state (smallest λ) is axisymmetric for $L/a \lesssim 1.67$ and nonaxisymmetric for $L/a \gtrsim 1.67$. Adapted from [17]

at large enough N to capture the axial modes present in the system (i.e. for an aspect ratio of 10, $N \geq 10$).

Figure 1.5b shows the first two physical roots for solutions with $m = 0$ and $m = 1$ symmetry for $L/a < 20$. Solutions with $m \geq 2$ symmetry are not shown as they always have larger λ values for a given aspect ratio [13]. Crossings between the lowest order $m = 0$ and $m = 1$ modes occur at $L/a = 1.67$ and $L/a = 4.14$ in agreement with references 13 and 14. Because the minimum energy state corresponds to state with the smallest λ, nonaxisymmetric fields are expected whenever aspect ratios increase above 1.67. In the limit of an infinitely long cylinder, $\lambda a \to 3.11$, $ka \to 1.234$, and the solution describes a Taylor double helix [15, 16]. The magnetic field topology of a constant λ force free-state with $L/a = 10$ is shown in figure 1.6. The structure is essentially helical, resembling a spheromak that has been elongated and twisted along the major radius.

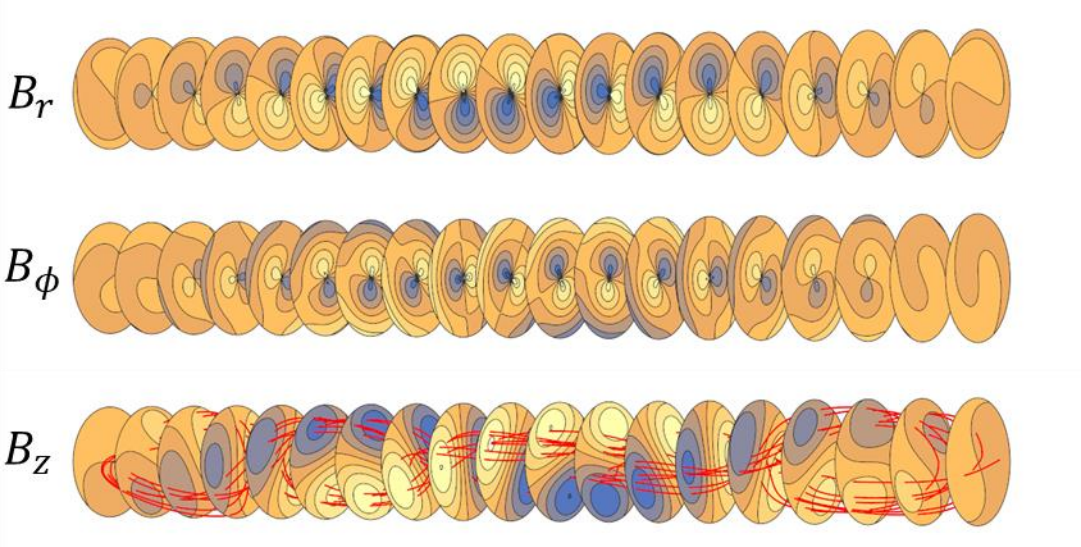


Figure 1.6: Isosurfaces of the lowest order $m = 1$ solution of a constant λ force-free state for $L/R = 10$. Field streamlines (red) have been included in the plot of isosurfaces for B_z. The structure is fundamentally helical, decaying in strength near the end wall at $z = 0$ and $z = L$.

1.4 Context: astrophysical and laboratory plasmas

While many space and laboratory plasmas are well approximated on a global scale using ideal MHD and the dynamics of magnetic flux tubes, deviations from ideal behavior at small scales can have an important effect on the dynamics of such systems [18-20]. The consequences of non-ideal behavior are generally thought to account for many longstanding mysteries in space and laboratory plasmas. A number of these are discussed below, providing a motivating context for the development of a generalized model that captures non-ideal effects while maintaining the intuition developed for ideal MHD plasmas.

1.4.1 Astrophysical Jets

Astrophysical jets [21] are collimated, high-speed outflows observed to be natural features of celestial objects that spin and accrete matter. From protoplanetary nebulae [22] to young stellar objects [23] and active galactic nuclei [24], common features suggest that universal mechanisms may be responsible for jet launching, collimation, and stability. Theoretical [25] and laboratory [26] hydrodynamic jet models suppose that polar funnels in a thick accretion disk behave as nozzles with high Mach number radiative flows responsible for collimation beyond the formation region [27]. These models however, cannot explain the collimation of very long jets observed from

low aspect ratio funnels [21], over-dense jets surrounded by vacuum or low-density medium [28], signatures of helical motion [29] or helical magnetic fields [30]. Therefore, theoretical [31] and laboratory [32] magnetohydrodynamic models include the presence and effect of magnetic fields for jet launching and collimation. A predominant model [31] holds that plasma in a differentially rotating accretion disk twists up initially poloidal ($\hat{r}, \hat{z}$) magnetic field (due to the frozen-in condition), generating a helical magnetic field structure. As the differential rotation continues, the magnetic pressure associated with the growing toroidal field leads to expansion in the axial direction. The outcome is a helical magnetic tower consisting of a highly wound toroidal field, which acts to collimate the jet. To provide collimation at high aspect ratios, some models require the presence of an external plasma medium; however, an MHD pumping mechanism [33] which convects toroidal magnetic flux from the central engine to pile up at the end tips of the jet has been proposed as an additional collimation mechanism. Experimental evidence for the MHD pumping mechanism has been observed in magnetized laboratory jet experiments produced by planar plasma guns designed to mimic an accretion disk around a magnetized central object [34, 35]. These laboratory jets attain aspect ratios of $\sim 10:1$ before experiencing ideal MHD current driven kink instabilities [34, 35]. Axial shear flows were predicted to stabilize such instabilities in a linear Z-pinch configuration with purely axial currents [36] and laboratory observations confirmed the theory by producing stable collimated jets with aspect ratios $> 30:1$ with fixed length, fixed end-points, and no poloidal (axial) magnetic field [37]. Both models are based on large electrical currents flowing within the jet, raising the question as to whether magnetic collimation and flow stabilization mechanisms can work together to produce high-aspect ratio jets with only one end tied to the central object. The theory of relative canonical helicity transport, outlined in chapter 2 of this thesis predicts that this may be accomplished through a dynamic reduction in magnetic helicity (i.e. unwinding of magnetic fields) driving stabilizing helical shear flows. The MoCHI.Labjet experiment, introduced in chapter 4, is designed to explore this possibility by producing sheared helical flows inside a driven, magnetized plasma jet launched from planar, concentric, azimuthally symmetric electrodes.

1.4.2 Coronal loops, magnetic clouds, and the coronal heating problem

The sun exhibits exceedingly complex dynamics leading to the formation and annihilation of magnetically active regions. These regions are often associated with violent and explosive solar

activity known as coronal mass ejections (CMEs) and solar flares. While solar flares represent a localized event in the atmosphere of the sun, CMEs are large scale eruptions of solar material and magnetic fields which can rise above the solar corona to be energetically ejected (fast CME) into the interplanetary medium or carried off by the prevailing solar wind (slow CME), thus becoming an interplanetary-CME (ICME).

In situe measurements of ICMEs led to the identification of a particular subclass of ejecta known as "magnetic clouds" (MCs) [38]. These MCs were defined at ~ 1 AU as interplanetary structures with a dimension on the order of 0.25 AU with enhanced magnetic field strength, smooth and large rotation of the magnetic field, and relatively low ion temperatures. While other large-scale solar wind structures can exhibit any of the above features, the combination of all these properties appears to be unique to magnetic clouds [39]. The magnetic field topology of MCs is often well described by self similarly expanding Lundquist solutions [40] or even double helix solutions with opposite axial flux components [41]. Statistically, MCs have been found to represent about one third of observed ICMEs, with this fraction varying with solar cycle from 15% at solar maximum to as much as 100% at solar minimum. [42]. It remains unknown why only a fraction of ICMEs are observed as magnetic clouds; however, possible explanations include observational selection effects, evolutionary phenomena in which simple MCs interact with each other to produce more complex ICMEs that do not retain the same properties, and distinct birth mechanisms in which non-cloud ICMEs are produced through a different process.

Although there is no broadly accepted theoretical mechanism behind CME formation, the interaction of large, arched magnetic structures in the solar corona likely plays an important role. These coronal loops vary in length from 10^6 to 10^9 m and have plasma velocities on the order of $5 - 30\ km\ s^{-1}$ and densities of $10^{15} - 10^{17}\ m^{-3}$ [43]. With Lundquist numbers of $\sim 10^{12}$ (indicative of highly conductive plasma) these loops are well described on a global scale by ideal MHD (which precludes dissipative phenomena such as magnetic reconnection). However, photospheric observations [44] show that reconnection must be taking place to prevent the infinite build-up of magnetic stress associated with the turbulent motion of the loop foot-points. This suggests the formation of small scale structures like current sheets or magnetic turbulence where ideal MHD is broken and magnetic diffusion and reconnection can occur quickly. These reconnection events convert stored magnetic energy into other forms of energy (i.e. thermal, kinetic, radiative) providing a possible source of energy for CMEs and solar flares. In the case of

small reconnection events (i.e. nanoflares [45]) the induced plasma motion is thought to occur at small scale lengths such that the imparted energy is soon dumped by turbulence and then by viscosity, providing heat to the plasma which can be conducted along magnetic field lines by free electrons. Provided the frequency and energy content of nanoflares is sufficient, they represent one candidate for the heating mechanism of the solar corona. By extending the concept of magnetic flux tubes and magnetic helicity transport to account for finite particle momentum, the theory of canonical helicity transport outlined in chapter 2 provides an important paradigm shift in the interpretation of reconnecting coronal loops. This new perspective may provide valuable insight into the mechanisms responsible for coronal heating, coronal mass ejections, and solar flares.

1.4.3 Merging experiments

Reconnection events within merging magnetic flux tubes have been observed to accelerate and heat ions [45]. The complex dynamics and final state of such processes appear to depend on the ion size parameter, a ratio of the characteristic system scale length L to the ion Larmor radius ρ_{Li} [46]. Because this parameter reveals the importance of ion kinetic and two-fluid effects, this suggests that the global plasma evolution during flux rope merging depends on effects not captured by the dynamics of reconnecting magnetic flux tubes under the constraint of constant magnetic helicity. In one such representative case [46], two counter helicity spheromaks merge to produce a single spheromak or field-reversed configuration (FRC) depending on whether the initial magnetic helicity of the system $\langle\lambda\rangle$ is above or below some critical value. Tests with varying gas types show that this threshold scales with the inverse of the ion size parameter [46].

While spheromaks are well predicted by relaxation theories based on the conservation of magnetic helicity (quasistatic, ideal-MHD), FRC's are not (strong flows, high β). However, from the perspective of a two-fluid flowing equilibria [47], FRCs correspond to a minimum energy state with finite flow helicity but no magnetic helicity. From this perspective it is apparent that there must be a conversion of the initial magnetic helicity into flow helicity (i.e. conversion of magnetic energy into kinetic energy). The theory of relative canonical helicity transport, outlined in chapter 2, is the first to predict that such transfers are possible via a generalized battery effect provided a non-equipotential enthalpy exists on surfaces common to both ion and electron canonical flux tubes. The model further explains why the choice of one path over the other depends on the scale lengths over which the enthalpy is imposed compared to the ion inertial scale length.

Chapter 2
A UNIFIED MODEL

2.1. Introduction

This chapter presents a unified, field-theory framework for the dynamics of single-particle, kinetic, and fluid plasma models derived directly from a Lagrangian-Hamiltonian formalism [48]. The framework results in a canonical form of the equation of motion that is valid across all regimes and reveals a more fundamental topological constraint than magnetic or fluid helicity. The model suggests that techniques borrowed from the theory of electromagnetism are suitable for the analysis of plasma dynamics and relaxation, while a generalized frozen-in condition extends intuition developed in the ideal MHD regime across all plasma regimes.

2.2. Definitions and equations

In Lagrangian mechanics, a canonical momentum is defined as the derivative of the Lagrangian with respect to a generalized velocity coordinate ($\partial\mathcal{L}/\partial\dot{q}$). In the case of a single particle traveling in a magnetic field, the appropriate Lagrangian can be shown to be

$$\mathcal{L}_\sigma = \frac{1}{2}m_\sigma v_\sigma^2 - q_\sigma V + q_\sigma \vec{v}\cdot\vec{A} \tag{2.1}$$

where V is the electrostatic potential and $\vec{A}$ is the magnetic vector potential (gravitational potential has been ignored). The corresponding canonical momentum is $\vec{P}_\sigma = m_\sigma\vec{v}_\sigma + q_\sigma\vec{A}$. The framework of Lagrangian mechanics is not exclusive to single particle systems however, and in reference 47, a generalized species Lagrangian is proposed which holds for single particle, kinetic, and fluid plasma regimes provided the appropriate definitions of species canonical momentum and enthalpy (h_σ) are used. Table 2.1 outlines the appropriate definitions in the non-relativistic limit. Full definitions, including relativistic effects can be found in table 1 of reference 48. In flat Minkowski space the generalized Lagrangian takes the form

$$\mathcal{L}_\sigma = \mathbb{q}_\sigma\vec{v}_\sigma\cdot\vec{P}_\sigma - \mathbb{q}_\sigma h_\sigma + \frac{1}{2}\epsilon_\sigma\Sigma_\sigma^2 - \frac{\Omega_\sigma^2}{2\mu_\sigma} \tag{2.2}$$

where $\mathbb{q}_\sigma$ is a generalized charge, ϵ_σ is a generalized permittivity, μ_σ is a generalized permeability, Σ_σ is a canonical force-field and Ω_σ is the canonical vorticity (Table 2.2). Inserting equation 2.2

into the Euler-Lagrange equation for the field $\vec{P}_\sigma$ results in a generalized set of Maxwell's equations that are valid in all plasma regimes:

Gen. Gauss' law: $$\nabla \cdot \vec{\Sigma}_\sigma = \frac{\mathbb{q}_\sigma}{\epsilon_\sigma} \tag{2.3}$$

Gen. Faraday's law: $$\nabla \times \vec{\Sigma}_\sigma = -\frac{\partial \vec{\Omega}_\sigma}{\partial t} \tag{2.4}$$

Gen. Gauss' law for magnetism: $$\nabla \cdot \vec{\Omega}_\sigma = 0 \tag{2.5}$$

Gen. Ampere's law $$\nabla \times \vec{\Omega}_\sigma = \mu_\sigma \vec{\mathbb{J}}_\sigma + \mu_\sigma \epsilon_\sigma \frac{\partial \vec{\Sigma}_\sigma}{\partial t}. \tag{2.6}$$

Using the definitions in Table 2.2, equation 2.3 can be recast in the form of a canonical equation of motion

$$\vec{\Sigma}_\sigma + \vec{v}_\sigma \times \vec{\Omega}_\sigma = \vec{R}_\sigma \tag{2.7}$$

which recovers the single particle equation of motion, the Boltzmann equation (1.1) for a kinetic group, and the fluid equations (1.4) provided the corresponding definitions from table 2.1 are used. Remarkably, equation 2.7 has the same form as Ohm's law (equation 1.10) where the electric field has been replaced by a canonical force field, the bulk velocity has been replaced by the species velocity, the magnetic field has been replaced by the canonical vorticity, and the electrical resistivity has been replaced by a generalized dissipation term.

Table 2.1
Definitions of Canonical Quantities in Each Regime*

Regime	Single Particle	Kinetic	Fluid
Canonical Momentum	$\vec{P}_\sigma^{par} = m_\sigma \vec{v}_\sigma + q_\sigma \vec{A}$	$\vec{P}_\sigma^{kin} = f_\sigma (m_\sigma \vec{v}_\sigma + q_\sigma \vec{A})$	$\vec{P}_\sigma^{flu} = \rho_\sigma \vec{u}_\sigma + \rho_{c\sigma} \vec{A}$
Enthalpy	$h_\sigma^{par} = \frac{1}{2} m_\sigma v_\sigma^2 + q_\sigma V + m_\sigma \phi_g$	$h_\sigma^{par} = f_\sigma \left(\frac{1}{2} m_\sigma v_\sigma^2 + q_\sigma V + m_\sigma \phi_g \right)$	$h_\sigma^{flu} = \frac{1}{2} \rho_\sigma u_\sigma^2 + \rho_{c\sigma} V + \mathcal{P}_\sigma + \rho_\sigma \phi_g$

*Full definitions including relativistic contributions can be found in Table 1 of reference 48.

Table 2.2
Field Quantity Definitions

Regime	Single Particle
Canonical Force Field	$\vec{\Sigma}_\sigma \equiv -\nabla h_\sigma - \frac{\partial \vec{P}_\sigma}{\partial t}$
Canonical Vorticity	$\vec{\Omega}_\sigma = \nabla \times \vec{P}_\sigma$
Gen. Permittivity	$\epsilon_\sigma \equiv \frac{\epsilon_0}{q_\sigma^2}\left(\frac{\epsilon_0}{f_\sigma^2 q_\sigma^2}\ kinetic\right)$
Gen. Permeability	$\mu_\sigma \equiv q_\sigma^2 \mu_0\ \ (f_\sigma^2 q_\sigma^2 \mu_0\ kinetic)$
Dissipative Forces	$\vec{R}_\sigma \equiv -\nabla(\frac{1}{2}\epsilon_\sigma \vec{\Sigma}_\sigma - \frac{\vec{\Omega}_\sigma}{2\mu_\sigma})$
Gen. Charge	$\mathbb{q}_\sigma \equiv \epsilon_\sigma \nabla \cdot \left(\vec{R}_\sigma - \vec{v}_\sigma \times \vec{\Omega}_\sigma\right) \equiv 1$

2.3. Canonical flux tubes and their helicity

Because of the isomorphism between equations 2.7 and 1.10, it is immediately evident that taking the curl of equation 2.7 in the absence of dissipative forces $\left(\vec{R}_\sigma = 0\right)$ results in a new frozen-in flux condition. In this case, the canonical vorticity flux

$$\Psi_\sigma = \oiint \vec{\Omega}_\sigma \cdot d\vec{S} \tag{2.8}$$

is frozen to the species velocity $\vec{v}_\sigma$ as opposed to the bulk plasma velocity. In the case of an axisymmetric system, this can also be demonstrated by relating the canonical vorticity flux to the canonical angular momentum

$$\Psi_\sigma = 2\pi r P_{\sigma\phi}. \tag{2.9}$$

Since the particle motion preserves angular momentum, surfaces of constant canonical vorticity flux represent particle drift surfaces. Therefore, tubes of constant canonical vorticity flux (canonical flux tubes for short) replace magnetic flux tubes as the primary building block of multifluid or kinetic plasmas, extending flux tube physics to include the effects of finite species momentum.

In the fluid plasma regime,

$$\vec{\Omega}_\sigma = n_\sigma m_\sigma \vec{\omega}_\sigma + n_\sigma q_\sigma \vec{B} + \frac{\nabla n_\sigma}{n_\sigma} \times \vec{P}_\sigma \tag{2.10}$$

where $\vec{\omega}_\sigma = \nabla \times \vec{u}_\sigma$ is the species flow vorticity. For uniform densities ($\nabla n_\sigma = 0$), canonical flux tubes can thus be expressed as a weighted sum of a magnetic flux $\psi = \oiint \vec{B} \cdot d\vec{S}$ and a flow vorticity flux $\mathcal{F}_\sigma = \oiint \vec{\omega} \cdot d\vec{S}$

$$\Psi_\sigma = n_\sigma m_\sigma \mathcal{F}_\sigma + n_\sigma q_\sigma \psi. \tag{2.11}$$

Dotting the curl of equation 2.7 with $\vec{\Omega}_\sigma$ gives

$$\nabla h_\sigma \cdot \vec{\Omega}_\sigma \tag{2.12}$$

in steady-state conditions without dissipation. This is a generalization of the magnetostatic $\nabla p \cdot \vec{B} = 0$ where the pressure gradient is always perpendicular to the magnetic flux surfaces. Supposing that light emission is roughly proportional to the square of the particle density in a uniform temperature ideal MHD plasma, astrophysical photographs would be signatures of magnetic flux surfaces. In the more general case of a flowing, barotropic plasma where enthalpy is purely density driven (so equation 2.12 becomes $\nabla n_\sigma \cdot \vec{\Omega}_\sigma = 0$), then photographs would be signatures of canonical vorticity flux surfaces, not magnetic flux surfaces.

As demonstrated in sections 1.3.2 – 1.3.3, plasma often appear to self-organize into certain preferred states, independent of the detailed dynamics but subject to constrained relaxation [49]. In the ideal MHD regime, magnetic helicity was demonstrated to be a rugged topological invariant that constrained relaxation towards force-free states with constant λ. However, plasmas are not generally force-free, and may exhibit significant fluid pressures and flows, so a model based on a more general helicity such as the species canonical helicity

$$K_\sigma \equiv \iiint \vec{P}_\sigma \cdot \vec{\Omega}_\sigma \, dV \tag{2.13}$$

(i.e. the helicity of canonical flux tubes) provides a more fundamental approach to helicity constrained relaxation. Such a model [50, 51] retrieves all the previous results for static ideal MHD plasmas [4, 6, 7, 52] and ordinary neutral fluids [53] in the appropriate limits, while extending the same topological concepts to regimes where flowing multi-fluid or kinetic plasmas are applicable [47, 48, 51, 54].

Like magnetic helicity, canonical helicity is only well-defined within isolated volumes with closed field lines. Early generalizations of helicity [51, 55-57] therefore ignored situations where canonical flux tubes intercepted the boundary of a system and, even within an isolated system, ignored that each species' canonical flux tubes can overlap and intercept one another. These treatments thus concluded that each species' canonical helicity was itself invariant ($\dot{K}_\sigma = 0$). The theory of relative canonical helicity transport [50] accounts for gauge dependence and generalizes the concepts of helicity evolution to open and driven configurations. The theory demonstrates that the species' canonical helicities are coupled, that the system's total helicity $\mathbb{K} \equiv \sum_\sigma K_\sigma$ is the appropriate system invariant, and that canonical helicity can be transferred between species.

Gauge-invariant canonical helicity is defined in a manner analogous to relative magnetic helicity (equation 1.16) [50]

$$K_{\sigma rel} \equiv \iiint \left(\vec{P}_\sigma - \vec{P}_{\sigma ref}\right) \cdot \left(\vec{\Omega}_\sigma + \vec{\Omega}_{\sigma ref}\right) dV \tag{2.14}$$

where $\vec{P}_\sigma$ and $\vec{\Omega}_\sigma$ are the canonical momentum and canonical vorticity, respectively, while $\vec{P}_{\sigma ref}$ and $\vec{\Omega}_{\sigma ref}$ are reference fields to be defined later. A small difference between this definition and the definition of relative magnetic helicity is the order with which the reference fields are added or subtracted. This difference arises from the fact that canonical helicity includes physical constants while magnetic helicity does not, and that the species enthalpy is naturally useful only as a relative quantity ($h_{\sigma-} \equiv h_\sigma - h_{\sigma ref}$); thus, $\vec{P}_{\sigma-} \equiv \vec{P}_\sigma - \vec{P}_{\sigma ref}$ must be used as opposed to $\vec{P}_{\sigma+} \equiv \vec{P}_\sigma + \vec{P}_{\sigma ref}$ even though both formulations are gauge invariant. To accommodate this change in sign, the reference fields must satisfy the conditions [58]

$$\begin{cases} \nabla \times \vec{\Omega}_{ref}\big|_V = 0 \\ \vec{\Omega}_\sigma \cdot \hat{n}|_S = -\vec{\Omega}_{\sigma ref} \cdot \hat{n}|_S \\ \nabla \times \vec{P}_{\sigma ref} = \vec{\Omega}_{ref} \end{cases} \tag{2.15}$$

i.e. the normal component of the reference potential field $\vec{\Omega}_{\sigma ref}$ must be opposite to that of $\vec{\Omega}_\sigma$ at the boundaries. A convenient expansion of equation (2.14) shows that the relative canonical helicity can be thought of as the weighted sum of three helicity components

$$K_{\sigma rel} = m_\sigma^2 \mathcal{H}_{\sigma rel} + m_\sigma q_\sigma \mathcal{X}_{\sigma rel} + q_\sigma^2 \mathcal{K}_{rel} \tag{2.16}$$

where

$$\mathcal{H}_{\sigma rel} = \iiint n_\sigma^2 \vec{u}_{\sigma-} \cdot \vec{\omega}_{\sigma+} dV \tag{2.17}$$

is the relative kinetic helicity,

$$\mathcal{X}_{\sigma rel} = \iiint n_\sigma^2 \left(\vec{u}_{\sigma-} \cdot \vec{B}_+ + \vec{u}_{\sigma+} \cdot \vec{B}_-\right) dV \tag{2.18}$$

is the relative cross helicity, and $\mathcal{K}_{rel}$ (defined in equation 1.16) is the relative magnetic helicity with the $\pm$ subscripts switched. Thus, while canonical flux tubes depend on density gradients, their canonical helicity is independent of any density gradients but is locally weighted by the square of the number density n_σ^2. For the case of massless electrons in a reduced two-fluid plasma, the electron canonical helicity is simply the magnetic helicity weighted by the charge density squared.

The evolution of relative canonical helicity in a volume is obtained by taking the derivative of equation 2.14 and using equation 2.7 (and its curl) giving

$$\begin{aligned} \frac{dK_{\sigma rel}}{dt} = & -2 \iiint \vec{R}_\sigma \cdot \vec{\Omega}_\sigma dV - 2 \oiint h_{\sigma ref} \vec{\Omega}_{\sigma ref} \cdot d\vec{S} \\ & - \iiint \frac{\partial \left(\vec{P}_{\sigma ref} \cdot \vec{\Omega}_{\sigma ref}\right)}{\partial t} dV - \oiint \vec{P}_{\sigma ref} \times \frac{\partial \vec{P}_{\sigma ref}}{\partial t} \cdot d\vec{S} \\ & + \oiint \left(\vec{P}_{\sigma-} \cdot \vec{\Omega}_{\sigma+}\right) U \cdot d\vec{S} \end{aligned} \tag{2.19}$$

where the terms on the right-hand side represent a dissipative term, a generalized battery term, a term due to changes of the reference fields inside the volume, an inductive term, and a Leibniz term due to the motion U of the boundary of the system, respectively. Considering helicity injection due to an enthalpy difference Δh_σ across the ends of a uniform density canonical flux tube in a fixed, canonical flux conserving volume (such that terms 3-5 on the right vanish and $h_{\sigma ref} \vec{\Omega}_{\sigma ref} \cdot d\vec{S} = h_\sigma \vec{\Omega}_\sigma \cdot d\vec{S}$) with no dissipation ($\vec{R}_\sigma = 0$) equation 2.19 can be written explicitly to distinguish between the flow vorticity tube component and the magnetic flux tube component

$$\dot{K}_{\sigma rel} = 2 n_\sigma \Delta h_\sigma (m_\sigma \mathcal{F}_\sigma + q_\sigma \psi) \tag{2.20}$$

where we have assumed a uniform relative enthalpy over the tube cross-sectional area. The ratio between the two terms on the right-hand side of equation 2.20 can be defined as a fractional canonical helicity injection threshold [50]

$$\overline{K}_{thr} \equiv \frac{|m_\sigma \Delta h_\sigma \mathcal{F}_\sigma|}{|q_\sigma \Delta h_\sigma \psi|} \sim \frac{m_\sigma u_{\sigma th}}{q_\sigma LB} \sim \frac{\rho_{\sigma L}}{L} \sim \frac{1}{S^*} \tag{2.21}$$

where the size parameter S^* represents the ratio of the system scale length L to the species Larmor radius $\rho_{\sigma L}$. For large size parameter, a given enthalpy difference applied to a canonical flux tube will preferentially channel helicity into the magnetic flux tube component, and for a small size parameter, the same enthalpy difference will channel helicity into the flow vorticity component. This result provides a first-principles explanation for why the bifurcation in compact-torus merging experiments depends on the ion size parameter [46]. Furthermore, the results demonstrate that even within an isolated volume, different species helicities can be coupled to each other when one species is taken as the reference for the other species. The total relative canonical helicity $\mathbb{K}_{rel} \equiv \sum_\sigma K_{\sigma rel}$ is conserved in the absence of dissipation and the coupling between the two species is represented by terms 2-4 on the right-hand side of equation 2.19, where the surface integrals are over common canonical vorticity surfaces.

From the point of view of a uniform density reduced two-fluid plasma ($m_e \to 0$), this transport model shows that magnetic helicity (now topologically equivalent to electron canonical helicity) can be transferred to ion kinetic helicity while preserving the total relative canonical helicity of the system. Physically, this topological argument would be manifested as an unwinding of magnetic fields driving helical flows, and vice versa, as a winding of magnetic fields driving axial flows. Conversion between one helicity and another would depend on the scale lengths (size parameter) where the enthalpy is applied.

The ruggedness of canonical helicity can be examined in an analogous manner to magnetic helicity (equation 1.20) where the total system energy is expressed in terms of the total Hamiltonian $\mathbb{H}_\sigma = \oiiint H_\sigma$ and the Hamiltonian density is defined as [48]

$$H_\sigma = \frac{1}{2}\epsilon_\sigma \Sigma_\sigma^2 + \frac{\Omega_\sigma^2}{2\mu_\sigma} + \mathbb{q}_\sigma h_\sigma \,. \tag{2.22}$$

In the simplest case of a fixed, isolated system that is subject to dissipative forces, the ratio of changes in species energy to species canonical helicity becomes

$$\frac{\Delta H_\sigma}{\Delta K_\sigma} = \frac{1}{2}\frac{\oiint \vec{R}_\sigma \cdot \vec{v}_\sigma \, dV}{\oiint \vec{R}_\sigma \cdot \vec{\Omega}_\sigma \, dV} \sim \frac{v_\sigma}{2\Omega_\sigma}. \tag{2.23}$$

In the fluid regime, the factor $v_\sigma/2\Omega_\sigma$ approximates an inverse density gradient scale length L_s/ρ_σ where the scale length $L_s \equiv L_{circ}\rho_{\sigma L}/2(L_{circ} + \rho_{\sigma L})$ is a hybrid scale length of the Larmor radius $\rho_{\sigma L}$ and the characteristic fluid vorticity scale length L_{circ}. In kinetic regimes the fluid density ρ_σ is replaced by $f_\sigma m_\sigma$ and the gradient of the distribution function plays an important role in the ruggedness of canonical helicity. Equation 2.23 shows that species canonical helicity is a rugged invariant for shallow density gradients. By minimizing the total magnetofluid energy subject to the constraint of fixed species canonical helicity, more general equilibria can be obtained [47, 51, 54]

$$\nabla \times \vec{\Omega}_\sigma = \lambda_\sigma \vec{\Omega}_\sigma \tag{2.24}$$

which is analogous to equation 1.23 and contains all the classical force-free solutions, while also permitting solutions with finite β and strong helical shear flows that resemble FRC and tokamak like equilibria.

Chapter 3
THE TOPOLOGY OF CANONICAL FLUX TUBES IN FLARED JET GEOMETRY

3.1. Introduction

Magnetized plasma jets are generally modeled as magnetic flux tubes filled with flowing plasma governed by magnetohydrodynamics (MHD). In this chapter, we investigate a more fundamental approach based on flux tubes of canonical vorticity. This approach extends the concepts of magnetic flux tube evolution to include the effects of finite particle momentum and enables visualization of the topology of plasma jets in regimes beyond MHD. A flared, current-carrying magnetic flux tube in an ion-electron plasma with finite ion momentum is thus equivalent to either a pair of electron and ion flow flux tubes, a pair of electron and ion canonical momentum flux tubes, or a pair of electron and ion canonical vorticity flux tubes. Here we examine the morphology of these flux tubes for increasing electrical currents, different radial current profiles, different electron Mach numbers, and a fixed, flared, axisymmetric magnetic geometry. These geometric and boundary conditions were chosen to be relevant to planar plasma gun laboratory setups designed to mimic an accretion disk-astrophysical jet system [32, 34]. These calculations intend to provide an intuitive, kinematic view of the evolution of jets without resorting to costly fully self-consistent dynamical calculations. The electrons are considered to be massless only to simplify the picture to just two flux tubes (one ion and one electron, equivalent to a magnetic flux tube) but can readily be extended for the more general case of three or more flux tubes (ions, electrons, and magnetic tubes). Calculations of gauge-invariant relative canonical helicities track the evolution of magnetic, cross, and kinetic helicities in the system and show that ion flow fields can unwind to compensate for an increasing magnetic twist. The results demonstrate that including a species' finite momentum can result in a very long, collimated canonical vorticity flux tube even if the magnetic flux tube is flared. With finite momentum, particle enthalpy (i.e. density) gradients must be normal to canonical vorticities (equation 2.12), not to magnetic fields, so observations of collimated astrophysical jets could be images of canonical vorticity flux tubes instead of magnetic flux tubes.

3.2. Visualization of canonical flux tubes

As opposed to magnetic flux tubes, the concept of canonical flux tubes is a little more abstract, so we develop a numerical visualization tool to examine the shapes of tubes and canonical fields for a flared magnetic flux tube carrying various profiles of flow and current density. The tool also calculates the relative canonical helicity content of the jet to examine the individual contribution from each helicity component to the total canonical helicity. This quasi-static visualization code is not designed to elucidate the full, self-consistent dynamics of canonical flux tubes, but to reveal the corresponding canonical viewpoint for a driven magnetic flux tube as current or fluid momentum is ramped up. This is similar to how the evolution of magnetic flux tubes in force-free plasmas are traced as current is ramped up [59], but with the important distinction that the canonical (and magnetic) fields used here are not force-free. For example, the evolution of a stable screw pinch or force-free state can be approximated by sweeping $\lambda = \mu_0 I/\psi$ values in equation 1.23 (I is the total current carried by the configuration, μ_0 is the permeability of free space, and ψ is the magnetic flux in the configuration) whereas a general magnetic flux tube can kinematically evolve through various non-force-free states as axial current is applied. It is in the latter manner that we vary the enthalpy boundary conditions (i.e. axial current profile and fluid momentum) to investigate the geometry and topology of canonical fields with respect to magnetic and fluid velocity fields.

Table 3.1
Poloidal Flux Functions and Flux Coordinates for Pertinent Vector Fields

Flux Coordinates	Vector Field General	Magnetic	Fluid (Incompressible)	Fluid Vorticity	Canonical Vorticity
$\hat{\boldsymbol{e}}_1$	$\nabla f/\lvert\nabla f\rvert$	$\widehat{\boldsymbol{\psi}}$	$\widehat{\boldsymbol{\psi}}_{\boldsymbol{\sigma}}$	$\widehat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{\sigma}}$	$\widehat{\boldsymbol{\Psi}}_{\boldsymbol{\sigma}}$
$\hat{\boldsymbol{e}}_2$	$\widehat{\boldsymbol{\theta}} = r\nabla\theta$	$\widehat{\boldsymbol{\theta}}$	$\widehat{\boldsymbol{\theta}}$	$\widehat{\boldsymbol{\theta}}$	$\widehat{\boldsymbol{\theta}}$
$\hat{\boldsymbol{e}}_3$	$\dfrac{\nabla f \times \nabla\theta}{\lvert\nabla f \times \nabla\theta\rvert}$	$\widehat{\boldsymbol{b}}_{\boldsymbol{pol}}$	$\widehat{\boldsymbol{u}}_{\boldsymbol{\sigma pol}}$	$\widehat{\boldsymbol{\omega}}_{\boldsymbol{\sigma pol}}$	$\widehat{\boldsymbol{\Omega}}_{\boldsymbol{\sigma pol}}$
Poloidal flux function	$f = \iint \boldsymbol{X} \cdot d\boldsymbol{S}$	$\boldsymbol{X} = \boldsymbol{B}$	$\boldsymbol{X} = \boldsymbol{u}_{\boldsymbol{\sigma}}$	$\boldsymbol{X} = \boldsymbol{\omega}_{\boldsymbol{\sigma}}$	$\boldsymbol{X} = \boldsymbol{\Omega}_{\boldsymbol{\sigma}}$

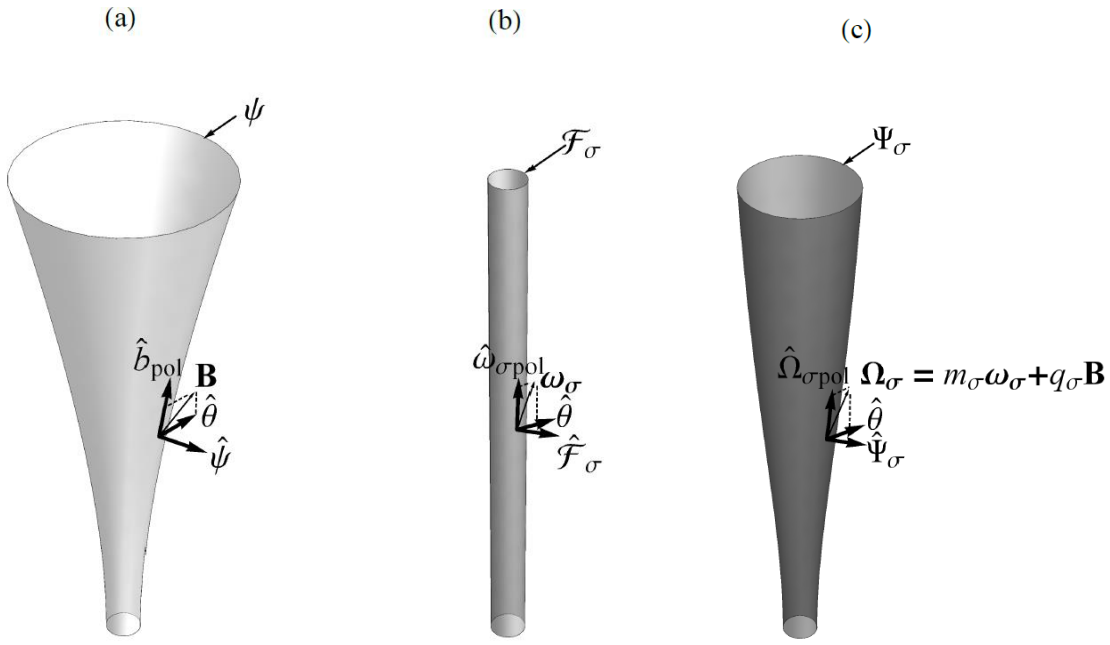


Figure 3.1 Example flux coordinate directions for (a) a magnetic flux tube ψ, (b) a flow vorticity flux tube $\mathcal{F}_\sigma$ of species σ, and (c) canonical vorticity flux tube Ψ_σ for the species σ. The directions are determined by the scalar flux functions (Table 1). The canonical vorticity flux coordinates in panel (c) are the weighted sum of the magnetic and flow vorticity components of panels (a) and (b).

For simplicity, the plasma jets are assumed to be axisymmetric. Any solenoidal vector field can therefore be decomposed into poloidal ($\hat{r}, \hat{z}$) and toroidal ($\hat{\theta}$) scalar flux functions (f and g respectively) such that a given vector field $\boldsymbol{X}$ can be written as

$$\boldsymbol{X} = \boldsymbol{\nabla} f \times \frac{\hat{\boldsymbol{\theta}}}{r} + g\frac{\hat{\boldsymbol{\theta}}}{r}. \tag{3.1}$$

Table 3.1 and figure 3.1 illustrate how a right-handed orthogonal coordinate system can be constructed based on the poloidal flux function f for a given vector field and applied to four of our system's vector fields (magnetic field, fluid flow field, fluid vorticity field, canonical vorticity field).

Furthermore, we assume that our plasma jet has a flared magnetic field and carries magnetic flux in a tube of finite length and no external flows. There are thus three possible free

parameters for which we choose the following [60]. The first free parameter is the poloidal (vacuum) magnetic field. To provide a reasonably realistic flaring shape, the poloidal magnetic field is determined by a hypothetical current loop located at $z = 0$ and oriented parallel to the z-axis such that the magnetic vector potential is given by the single-turn current loop solution

$$A_\theta(r,z) = \frac{\mu_0 I_{\mathrm{vac}} a_{\mathrm{vac}}}{\pi\sqrt{a_{\mathrm{vac}}^2 + r^2 + z^2 + 2ra_{\mathrm{vac}}}}\left(\frac{(2-k^2)K(k^2) - 2E(k^2)}{k^2}\right) \tag{3.2}$$

where $k^2 = 4\,r\,a_{vac}/(a_{vac}^2 + r^2 + z^2 + 2\,r\,a_{vac})$, I_{vac} and a_{vac} are the ring current and radius, respectively and $K(k^2)$ and $E(k^2)$ are the complete elliptic integrals of the first and second kind. Because of azimuthal symmetry, equation 3.2 can be used to obtain the poloidal magnetic flux function $\psi = 2\pi r A_\theta$ and the poloidal magnetic field

$$\boldsymbol{B}_{\mathbf{pol}} = \frac{1}{2\pi}\left(\boldsymbol{\nabla}\psi \times \frac{\widehat{\boldsymbol{\theta}}}{r}\right). \tag{3.3}$$

The jet is radially bounded by a magnetic flux surface ψ_{jet}, where $0 < \psi_{jet} < \psi_{loop}$ and ψ_{loop} is the total magnetic flux through the hypothetical current loop. This defines a magnetic flux tube with an axially varying radius $a(z)$, i.e. the flare. The jet length is defined as l and chosen to be several times the jet radius at $z = 0$.

The second free parameter of the system is the current density $\boldsymbol{j}$. Since the system is fixed with no time-changing magnetic flux, we can suppose that j_θ is zero and the current is purely poloidal. The shape of the current profile can be defined arbitrarily and its effect on the canonical fields is a principal focus of this investigation. Assuming the current flows along the magnetic flux surface, i.e. $\boldsymbol{j}_{\mathbf{pol}}(\psi)$, and $\boldsymbol{\nabla}\cdot\boldsymbol{j}_{\mathbf{pol}} = 0$ we assume that the profile is self-similar in the axial direction and introduce a normalized radial coordinate $\eta(r,z) = r/a(z)$ to build a function that describes the radial current profile $j_z(\eta)$. We examine four profiles: a uniform current profile, a diffuse core current profile, a diffuse-core-with-skin current, and a skin-only current profile (figure 3.2). Defining a total jet current I_{jet}, the axial current density as a function of radial and axial position can be solved from

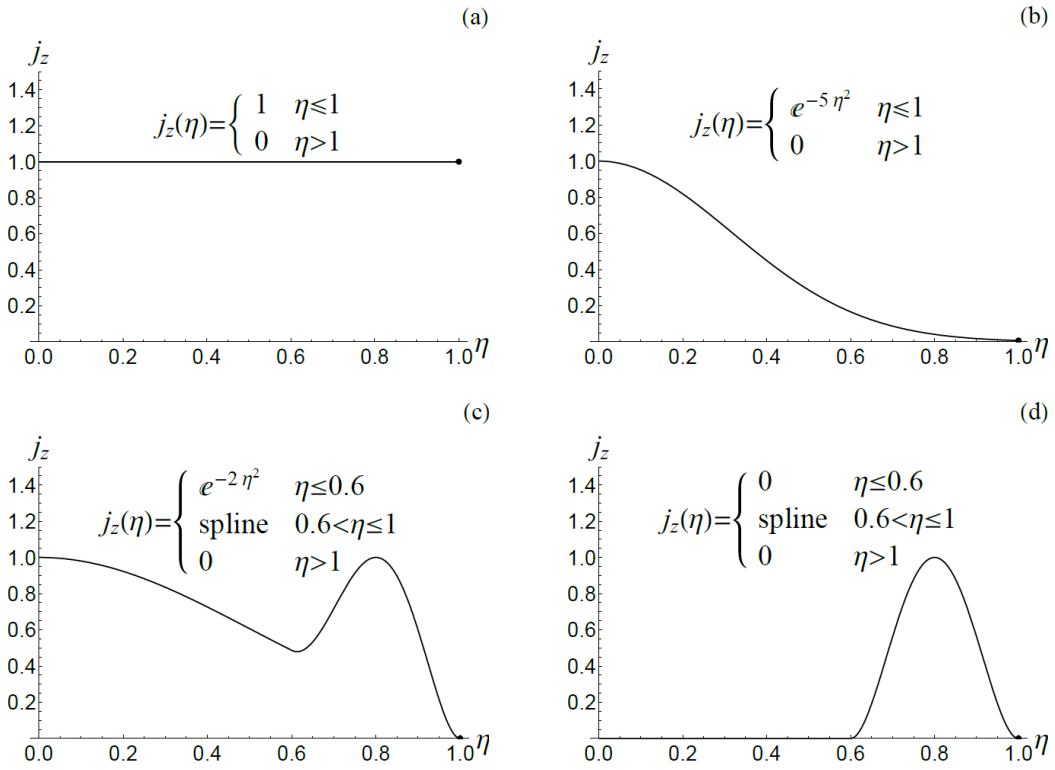


Figure 3.2. Shape of the axial current density j_z as a function of normalized radius η for (a) the uniform current profile, (b) the diffuse core current profile, (c) the diffuse-core-with-skin, and (d) skin-only current profile. The splines used in (c) and (d) are fifth-order interpolating polynomials set to match the value and slope of the core current at $\eta = 0.6$ with a maximum at $\eta = 0.8$ and going smoothly to zero at $\eta = 1$. These profiles are scaled by a multiplicative constant $j_0(z)$ to retrieve the total jet current I_{jet} when integrated over the cross-sectional area at any axial location (Equation (3.4)).

$$\begin{aligned} I_{\text{jet}} &= 2\pi \int_0^{a(z)} j_z(\eta) j_0(z) r dr \\ &= 2\pi \int_0^{a(z)} j_z(r,z) r dr \end{aligned} \tag{3.4}$$

where $j_0(z)$ is a scaling factor that depends on I_{jet} and the axial position. From here, the radial current density j_r can be inferred and the toroidal magnetic field calculated with Ampere's law

$$B_\theta(r,z) = \frac{\mu_0 I(r,z)}{2\pi r}. \tag{3.5}$$

The final free parameter is the magnitude of the electron flow velocity. Because we are considering a reduced two-fluid where electrons are massless, electrons are tied to the magnetic field lines (i.e. $\boldsymbol{u_e} = u_e\widehat{\boldsymbol{b}}$, where $\widehat{\boldsymbol{b}} = \boldsymbol{B}/|\boldsymbol{B}|$)with a velocity determined, in principle, by some parallel electric field or pressure gradients in equation 2.7. For this analysis, we will choose a constant as a first approximation. Assuming a uniform density plasma in which $n_i = n_e = n_0$, the current density and electron flow velocity then implicitly define the ion fluid velocity as

$$\boldsymbol{u_i} = \frac{1}{Z}\left(\frac{\boldsymbol{j}}{n_0 e} + \boldsymbol{u_e}\right) \tag{3.6}$$

where Z is the degree of ionization. Flow vorticities are obtained by taking the circulation of the velocity fields $\boldsymbol{\omega_\sigma} = \boldsymbol{\nabla} \times \boldsymbol{u_\sigma}$, and the poloidal component of the magnetic vector potential $\boldsymbol{A_{pol}} = \mu_0/(4\pi) \iiint(\boldsymbol{j}\, d^3\boldsymbol{r}'\, /|\boldsymbol{r} - \boldsymbol{r}'|)$ is found by integrating the current density over the jet volume.

With these field components defined, the canonical fields can be constructed for the electrons and the ions as follows:

$$\begin{cases} \boldsymbol{P_e} = -e\boldsymbol{A} \\ \boldsymbol{\Omega_e} = -e\boldsymbol{B} \end{cases} \qquad \begin{cases} \boldsymbol{P_i} = m_i\boldsymbol{u_i} + Ze\boldsymbol{A} \\ \boldsymbol{\Omega_i} = m_i\boldsymbol{\omega_i} + Ze\boldsymbol{B} \end{cases} \tag{3.7}$$

where the density component has been ignored since it is uniform throughout the volume. These vector fields are converted from cylindrical to Cartesian coordinates, and their streamlines are plotted parametrically by solving the equation

$$\frac{\partial \boldsymbol{r}(s)}{\partial s} = \boldsymbol{X}(\boldsymbol{r}(s)) \tag{3.8}$$

where s is a parametric index, $\boldsymbol{r}(s)$ is the position vector, and $\boldsymbol{X}$ is the vector field.. Canonical vorticity flux surfaces are calculated in the same manner as the magnetic flux surface by setting the canonical vorticity flux to be a constant, in this case $\Psi_\sigma = 2\pi a_0 P_{\sigma\theta}(a_0, 0)$, i.e. the total canonical vorticity flux through the jet radius at $z = 0$. For massless electrons $\Psi_e = -e\psi$ and the electron canonical vorticity flux surfaces are indistinguishable from magnetic flux surfaces. For the ions however, finite momentum can lead to significant departures of the ion canonical vorticity flux surfaces from the magnetic flux surfaces.

For flexibility, we normalize our quantities and governing equations by $\bar{\boldsymbol{\nabla}} = l_0\boldsymbol{\nabla}$, $\bar{\boldsymbol{B}} = \boldsymbol{B}/B_0$, $\bar{q}_\sigma = q_\sigma/e$, $\bar{m}_\sigma = m_\sigma/m_i$, $\bar{\boldsymbol{A}} = \boldsymbol{A}/(B_0 l_0)$, $\bar{\boldsymbol{u}}_\sigma = \boldsymbol{u_\sigma}/v_a$, $\bar{t} = tv_a/l_0$, $\bar{\phi} = \phi e/(m_i v_a^2)$, $\bar{p}_\sigma = 2\mu_0 p_\sigma/B_0^2$, where $v_a = B_0/\sqrt{\mu_0 n_0 m_i}$. Here l_0 is a convenient system scale length B_0 is a

convenient scale magnetic field strength. Neglecting friction, the normalized canonical equation of motion (equation 2.7) becomes

$$\frac{\partial \bar{\boldsymbol{P}}_{\boldsymbol{\sigma}}}{\partial \bar{t}} - \bar{\boldsymbol{u}}_{\sigma} \times \bar{\boldsymbol{\Omega}}_{\sigma} = -\bar{\boldsymbol{\nabla}} \bar{h}_{\sigma} \tag{3.9}$$

with the normalized canonical fields

$$\begin{cases} \bar{\boldsymbol{P}}_{\boldsymbol{\sigma}} = \bar{m}_{\sigma} \bar{\boldsymbol{u}}_{\boldsymbol{\sigma}} + S_0 \bar{q}_{\sigma} \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{\Omega}}_{\boldsymbol{\sigma}} = \bar{m}_{\sigma} \bar{\boldsymbol{\omega}}_{\boldsymbol{\sigma}} + S_0 \bar{q}_{\sigma} \bar{\boldsymbol{B}} \\ \bar{h}_{\sigma} = \frac{1}{2} \bar{m}_{\sigma} \bar{u}_{\sigma}^2 + \bar{q}_{\sigma} \beta_{ES} + \frac{1}{2} \beta_{\sigma} \end{cases} \tag{3.10}$$

The parameter $S_0 = eB_0 l_0/(m_i v_a) = l_0/\rho_{L0}$ is a size parameter for an Alfvénic ion travelling in the characteristic magnetic field B_0; the parameter $\beta_{ES} = e\phi/m_i v_a^2$ is an electrostatic beta expressing the relative strength of the electrostatic field to the kinetic energy; and the parameter $\beta_{\sigma} = 2\mu_0 p_{\sigma}/B_0^2$ is the usual species beta expressing the relative strength of fluid pressure to magnetic pressure. Eliminating the magnetic field in equation (3.10) gives the ion canonical vorticity as

$$\bar{\boldsymbol{\Omega}}_{\boldsymbol{i}} = \bar{\boldsymbol{\omega}}_{\boldsymbol{i}} + \bar{m}_e \bar{\boldsymbol{\omega}}_{\boldsymbol{e}} - \bar{\boldsymbol{\Omega}}_{\boldsymbol{e}} \tag{3.11}$$

which shows how canonical quantities for each species are coupled to each other through the magnetic field, i.e. the ion canonical vorticity is antiparallel to the electron canonical vorticity with correction factors based on the strength of the species' flow vorticities.

The dimensionless relative canonical helicity is defined from the normalized canonical fields as

$$\bar{K}_{\sigma \mathrm{rel}} \equiv \iiint \bar{\boldsymbol{P}}_{\boldsymbol{\sigma-}} \cdot \bar{\boldsymbol{\Omega}}_{\boldsymbol{\sigma+}} \, d\bar{V} \tag{3.12}$$

where the normal component of the reference potential field $\bar{\boldsymbol{\Omega}}_{\boldsymbol{\sigma ref}}$ must be opposite that of $\bar{\boldsymbol{\Omega}}_{\boldsymbol{\sigma}}$ on the normalized system boundary (equation 2.15). Assigning the integration volume to be the driven magnetic flux tube, this boundary condition can be broken up into magnetic and flow vorticity components. This allows for the calculation of the normalized helicity components

$$K_{\sigma rel}^* \equiv m_{\sigma}^* \mathcal{H}_{\sigma rel}^* + m_{\sigma}^* q_{\sigma}^* S_0 \mathcal{X}_{\sigma rel}^* + {q_{\sigma}^*}^2 S_0^* \mathcal{K}_{rel}^* \tag{3.13}$$

as outlined in equation 2.16. Because the magnetic field lines are tangential to the side boundary of the integration volume, the only conditions on the reference magnetic fields are

$$\begin{cases} \boldsymbol{\nabla} \times \overline{\boldsymbol{B}}_{\mathbf{ref}}|_{V} = 0 \\ \overline{\boldsymbol{B}}_{\mathbf{ref}} \cdot \hat{\boldsymbol{z}}|_{\bar{l}} = -\overline{\boldsymbol{B}} \cdot \hat{\boldsymbol{z}}|_{\bar{l}} \\ \overline{\boldsymbol{B}}_{\mathbf{ref}} \cdot (-\hat{\boldsymbol{z}})|_{\mathbf{0}} = -\overline{\boldsymbol{B}} \cdot (-\hat{\boldsymbol{z}})|_{\mathbf{0}}. \end{cases} \tag{3.14}$$

This condition is satisfied by letting $\overline{\boldsymbol{B}}_{\mathbf{ref}} = -\overline{\boldsymbol{B}}_{\mathbf{pol}}$ since the poloidal magnetic field is a potential (vacuum) field that satisfies these boundary conditions. The reference magnetic vector potential is then $\overline{\boldsymbol{A}}_{\mathbf{ref}} = -\bar{A}_{\theta}\hat{\boldsymbol{\theta}}$. The flow vorticity field on the other hand is not always tangential to the side surface of the magnetic flux tube, so the correct reference field must be solved from Laplace's equation

$$\begin{cases} \nabla^2 \phi_{\sigma} = 0 \\ \boldsymbol{\nabla}\phi_{\sigma} \cdot \hat{\boldsymbol{n}}|_{\boldsymbol{s}} = -\overline{\boldsymbol{\omega}}_{\boldsymbol{\sigma}\mathbf{ref}} \cdot \hat{\boldsymbol{n}}|_{\boldsymbol{s}} \end{cases} \tag{3.15}$$

where $\boldsymbol{\nabla}\phi_{\sigma} = \overline{\boldsymbol{\omega}}_{\boldsymbol{\sigma}\mathbf{ref}}$ and $\hat{\boldsymbol{n}}$ is the surface normal. The solution to equation 3.15 is not unique up to a constant, but because we are only interested in the gradient of ϕ_{σ} it is sufficient for defining our reference field. The corresponding flow reference field is obtained from axisymmetry using $\overline{\boldsymbol{u}}_{\boldsymbol{\sigma}\mathbf{ref}} = \bar{\mathcal{F}}_{\sigma\mathrm{ref}}/2\pi\bar{r}\ \hat{\boldsymbol{\theta}}$ where $\bar{\mathcal{F}}_{\sigma ref}$ is the reference flow vorticity flux. The reference canonical fields can then be constructed from the weighted contribution of the magnetic and flow reference field components.

Figure 3.3 shows, for the case of a jet with a uniform radial current profile, the five equivalent viewpoints of the concept of a flux tube in a magnetized flowing plasma. The current-carrying magnetic flux tube $\psi \equiv \int \boldsymbol{B} \cdot d\boldsymbol{S}$ (figure 3.3a) is flared with twisted magnetic field lines due to the superposition of a poloidal current with a poloidal magnetic dipole field. The center of mass velocity $\boldsymbol{U}$ (equivalent to the ion velocity in a reduced two-fluid where $m_e = 0$) is stuck to the magnetic flux surfaces but not the field lines. Figure 3.3b shows the same system as a pair of electron and ion flow tubes $\psi_{\sigma} \equiv \int \boldsymbol{u}_{\boldsymbol{\sigma}} \cdot d\boldsymbol{S}$. The electron flow tube is geometrically coincident ($\Leftrightarrow$) with the magnetic flux tube since the electrons are here taken to be massless and thus flow along magnetic field lines at a constant velocity. The ions must compensate for the electron direction to carry the poloidal current, so the helical ion flow streamlines have a steeper pitch angle $\theta = \tan^{-1}(u_{iz}^{*}/u_{i\theta}^{*})$ than the helical electron flow streamlines. Since the ion flow streamlines also remain on the magnetic flux surface, the ion and electron flow tubes both coincide with the

magnetic flux tube. Figure 3.3c shows the same system as a pair of electron and ion flow vorticity flux tubes $\mathcal{F}_\sigma \equiv \int \boldsymbol{\omega_\sigma} \cdot d\boldsymbol{S}$. Because $j_\theta = 0$, both species have the same radial gradient in toroidal velocity and thus share identical axial flow vorticity $\omega_{\sigma z} = r^{-1}\partial/\partial r(r u_{\sigma\theta})$. At moderate to large jet currents, $\omega_{\sigma z}$ dominates the poloidal fluid vorticity and ion and electron vorticity flux tubes are nearly identical. Since the jet is flared, a toroidal fluid vorticity $\omega_{\sigma\theta} = \partial u_{\sigma r}/\partial z - \partial u_{\sigma z}/\partial r$ gives rise to helical streamlines that have slightly different pitch angles for each species. Figure 3.3d shows the same system as a pair of canonical momentum flux tubes $\mathcal{P}_\sigma \equiv \int \boldsymbol{P_\sigma} \cdot d\boldsymbol{S}$. The streamlines are again helical with a pitch angle that increases in the axial direction as the toroidal magnetic vector potential A_θ decreases far from the axis. The flux surfaces are distinct for each species since a finite ion momentum pulls ion canonical momentum flux surfaces closer to magnetic flux surfaces and keeps streamlines more twisted further up the tube (provided the ion flow is helical). Figure 3.3e shows the same system as a pair of canonical vorticity flux tubes $\Psi_\sigma \equiv \oiint \vec{\Omega}_\sigma \cdot d\vec{S}$. The electron canonical vorticity is antiparallel to the magnetic field and therefore geometrically coincident to the magnetic and flow velocity tubes. In contrast, the ion canonical vorticity flux tube departs from the magnetic flux surface due to the added contribution of the poloidal ion flow vorticity. All of these equivalent viewpoints range from the familiar current-carrying magnetic flux tubes to the more abstract pair of canonical flux tubes.

The strength of the canonical flux tube viewpoint (figure 3.3e) compared to the current-carrying magnetic flux tube viewpoint (figure 3.3a) is the ability to visualize behavior more general than permitted by MHD [60] while retaining our intuition of changing geometry: for example, laboratory observations show that reconnecting magnetic flux tubes can generate new magnetic flux tubes and generate strong flows due to two-fluid or kinetic effects [61]. Magnetic helicity conservation is based on MHD, and thus can only predict the topological evolution of magnetic flux tubes into other magnetic flux tubes but not into or from flows. On the other hand, canonical helicity conservation can predict the topological evolution of both magnetic and flow fields as linkage between magnetic and flow flux tubes [47, 50, 51, 54, 60]. Since enthalpy gradients at the boundaries can increase magnetic twist or generate helical flows depending on the uniformity of the enthalpy gradient (equations 2.19, 2.20), the next section investigates the kinematic evolution of canonical flux tubes with four different enthalpy conditions at the boundaries for the case of a flared, cylindrically symmetric geometry with massless electrons.

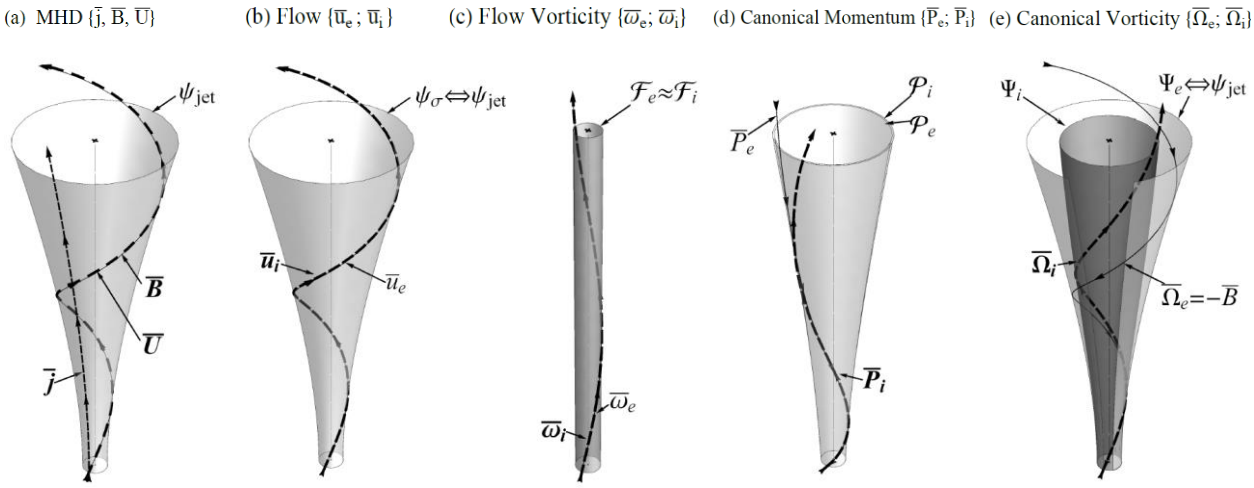


Figure 3.3 Five equivalent interpretations of flux tubes from the most familiar to the most abstract: (a) a current-carrying magnetic flux tube ψ_{jet} with uniform current profile and bulk flow velocity $\bar{\boldsymbol{U}}$ for $l_0^{-1}\lambda \approx 2.5$ and $|\bar{\boldsymbol{u}}_{\boldsymbol{e}}| = 3$; (b) ion and electron fluid velocity flux tubes ψ_σ with flows restricted to magnetic flux surfaces; (c) ion and electron fluid vorticity flux tubes $\mathcal{F}_\sigma$; (d) ion and electron canonical momentum flux tubes $\mathcal{P}_\sigma$; (e) ion and electron canonical vorticity flux tubes Ψ_σ. Here, the electron flux tube is coincident with the magnetic flux tube (light gray in panels a, b, and e) because of negligible electron mass while the ion flux tube of interest is highlighted (dark gray in panel e).

3.3. Kinematic evolution of flared plasma jet

The enthalpy driving the evolution of the magnetic flux tube (Figure 3.3a) and the canonical flux tube (figure 3.3e) is chosen to be the plasma jet current expressed as the dimensionless current parameter $\bar{\lambda} = \mu_0 I_{\text{jet}}/\psi_{\text{jet}} l_0$ where μ_0 is the permittivity of free space, I_{jet} is the total current inside the jet, ψ_{jet} is the total (fixed) magnetic flux, and l_0 is the characteristic scale length. The enthalpy is distributed across the cross-section of the tube as one of four possible current profiles (Figure 2) and either a sub- or super-Alfvénic electron flow velocity $|\bar{\boldsymbol{u}}_{\boldsymbol{e}}|$ (Table 3.2). Section 3.3.1 presents the cases with uniform and diffuse core currents. Section 3.3.2 presents the cases with diffuse-core-with-skin and skin-only currents. For all cases, the system constants are set to $m_i = m_{\text{proton}}$, $Z = 1$, $I_{\text{vac}} = 10^4$ A, $a_{\text{vac}} = 5$ m, $n_0 = 10^{20}\ m^{-3}$, $l = 15$ m, and $\psi_{\text{jet}} = 1$ mWb. The normalization values are chosen to be the magnetic field strength on axis $B_0 \sim 1.25$ mT and the jet radius at the footpoint $l_0 \sim 0.5$ m; resulting in an Alfvèn velocity of 2.74 $km\ s^{-1}$ and a size parameter of $S_0 \approx 22$. To illustrate shear between flux surfaces, streamlines are also presented on

Table 3.2
Selected Jet Parameters for Kinematic Evolution Calculations and Corresponding Figures

Enthalpy (Current Profile, Fig. 2)	**Electron Velocity: $\|\bar{u}_e\|$**	**MHD: $\{J\ ;\ B;\ U\}$** $\bar{\lambda}=1$	$\bar{\lambda}=10$	**Canonical: $\{\Omega_e\ ;\ \Omega_i\}$** $\bar{\lambda}=1$	$\bar{\lambda}=10$
Uniform	$\|\bar{u}_e\|=2/3$	Fig. 3.4a	Fig. 3.4b	Fig. 3.5a	Fig. 3.5b
	$\|\bar{u}_e\|=4$	Fig. 3.4c	Fig. 3.4d	Fig. 3.5c	Fig. 3.5d
Diffuse Core	$\|\bar{u}_e\|=2/3$	Fig. 3.4e	Fig. 3.4f	Fig. 3.5e	Fig. 3.5f
	$\|\bar{u}_e\|=4$	Fig. 3.4g	Fig. 3.4h	Fig. 3.5g	Fig. 3.5h
Diffuse Core with Skin	$\|\bar{u}_e\|=2/3$	Fig. 3.8a	Fig. 3.8b	Fig. 3.9a	Fig. 3.9b
	$\|\bar{u}_e\|=4$	Fig. 3.8c	Fig. 3.8d	Fig. 3.9c	Fig. 3.9d
Skin	$\|\bar{u}_e\|=2/3$	Fig. 3.8e	Fig. 3.8f	Fig. 3.9e	Fig. 3.9f
	$\|\bar{u}_e\|=4$	Fig. 3.8g	Fig. 3.8h	Fig. 3.9g	Fig. 3.9h

an interior magnetic ($\psi_{\text{inner}}=0.25$ mWb) and ion canonical flux tube ($\Psi_{i,inner}$) both defined to have the same radius at the starting footpoint $z=0$.

3.3.1. Uniform and diffuse current profiles

Figure 3.4 presents the morphology of a flared plasma jet with uniform and diffuse core current profiles from the MHD perspective of current-carrying magnetic flux tubes at low $\bar{\lambda}=1$ and high $\bar{\lambda}=10$ with a bulk velocity $\bar{\boldsymbol{U}}$ also effected by the choice of $|\bar{u}_e|$. In both uniform and diffuse current cases, as $\bar{\lambda}$ (i.e. jet current) is increased, the magnetic field becomes helical, with a shear in the non-uniform current case, due to the superposition of increasing poloidal current with a constant poloidal magnetic field. The bulk flow velocity $\bar{U}$ (equivalent to the ion flow velocity) is tied to magnetic flux surfaces with a steeper pitch angle than the magnetic field in regions of finite current density. This pitch angle is steeper with lower electron velocities since the ions must unwind to accommodate the poloidal current density. Because $j_\theta=0$, the magnetic flux surfaces remain unchanged as current is increased. However, from a canonical flux tube point-of-view (Figure 3.5), the shape of canonical flux surfaces depends on the species flow velocity as well as on the jet current profile and magnitude (Figure 3.6). Figures 3.5(a)-(d) show the equivalent systems to Figures 3.4(a)-(b) (uniform current profile) for a choice of $|\boldsymbol{u}_e|$ that is sub-Alfvénic (Figures 3.5(a)-(b)) or super-Alfévnic (Figures 3.5(c)-(d)). Figures 3.5(e)-(h) show the equivalent systems to Figures 3.4(c)-(d) (diffuse core current profile) for the same electron flow velocities. In

all cases, the electron canonical flux tubes are geometrically coincident with the magnetic flux tubes, $\Psi_e \Leftrightarrow \psi_{jet}$, because of our massless electron assumption. In all cases, the ion canonical flux tube departs from the magnetic flux surfaces $\Psi_i \nLeftrightarrow \psi_{jet}$ and becomes collimated further up away from the footpoint even though the initial cross-sectional area at the footpoint is the same as the magnetic flux tube. The collimation of the ion canonical flux tube occurs closer to the footpoint at larger currents $\bar{\lambda}$ (e.g. Figure 3.5(b) versus Figure 3.5(a)) and larger electron flow velocities $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ (e.g. Figure 3.5(d) versus Figure 3.5(b), and Figure 3.6).

Remembering that for an axisymmetric system the curl of a poloidal field is a toroidal field and vice versa, the separation of canonical flux surfaces can be explained as follows. At zero jet current, the ion flow has a small but finite toroidal circulation $\omega_{i\theta}$ because the ions flow with a constant velocity along flared magnetic field lines. Because the ion flow is purely poloidal at this point, then $\overline{\boldsymbol{\omega}}_{\mathbf{ipol}} = \mathbf{0}$ and the ion canonical vorticity flux is just the charge weighted magnetic flux $\Psi_i = q_i\psi$. With a finite jet current, the ion flows become helical and $\overline{\boldsymbol{\omega}}_{\mathbf{ipol}}$ becomes finite so the poloidal component of the ion canonical vorticity is no longer parallel to the poloidal magnetic field $\overline{\boldsymbol{\Omega}}_{\mathbf{ipol}} = \overline{\boldsymbol{\omega}}_{\mathbf{ipol}} + \overline{\boldsymbol{B}}_{\mathbf{pol}}$, causing the departure from the magnetic flux surfaces of the ion canonical vorticity flux surfaces (Figures 3.5 and 3.6).

The variation of poloidal flow vorticity $\overline{\boldsymbol{\omega}}_{\mathbf{ipol}} = \bar{\omega}_{ir}\hat{\boldsymbol{r}} + \bar{\omega}_{iz}\hat{\boldsymbol{z}}$ in space, with $\bar{\lambda}$, and with $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ results in the collimation of the canonical flux tube (Figure 6). The ion canonical flux tube is fully collimated when the radius of the canonical vorticity flux tube $a_{\Psi_i}(z)\sim$constant. i.e. when

$$\begin{cases} \bar{\omega}_{iz} \gg |\overline{\boldsymbol{B}}_{\mathrm{pol}}| \\ \bar{\omega}_{iz} \gg \bar{\omega}_{ir} \end{cases} \tag{3.16}$$

because $\overline{\boldsymbol{\Omega}}_{\mathbf{ipol}} = \overline{\boldsymbol{\omega}}_{\mathbf{ipol}} + \overline{\boldsymbol{B}}_{\mathbf{pol}}$. The ion canonical vorticity vector points in the direction of the (flared) poloidal magnetic field modified by the poloidal flow vorticity, which can cancel the flare if equation (27) is satisfied. Because the magnitude of ω_{iz} depends on the rate at which rB_θ (and thus $ru_{e\theta}$ and $ru_{i\theta}$) increases in the radial direction, the axial component of the flow vorticity is proportional to the current profile, i.e. $\omega_{iz} \propto \partial I(r,z)/\partial r$ where $I(r,z)$ is the total current at a given radial and axial location. Since the total current increases with radius for both current profiles, the axial component of the flow vorticity $\omega_{iz} = 1/r\ \partial(ru_{i\theta})/\partial r$ is always positive so always points up the tube. The radial profile $\omega_{iz}(r)$ is dictated by the shape and strength of the

current profile $j_z(r)$ and the magnitude of the electron flow velocity (curl of equation (3.6)). The radial component of the ion flow vorticity $\omega_{ir} = -\partial u_{i\theta}/\partial z$ is a consequence of the flaring of the jet and can be positive (point outward) or negative (point inward) at any given radial and axial location based on the shape of the current profile. For the uniform current cases of Figure 5, $\partial u_{i\theta}/\partial z \sim 0$ so $\omega_{ir} \ll \omega_{iz}$ and can be neglected. The ion canonical vorticity is then dominated by the sum of the poloidal magnetic field and axial flow vorticity, $\bar{\boldsymbol{\Omega}}_{\mathbf{ipol}} \simeq \bar{\omega}_{iz}\hat{\boldsymbol{z}} + \bar{\boldsymbol{B}}_{\mathbf{pol}}$. Close to the footpoint, the ion canonical vorticity departs little from the magnetic field because $\bar{\boldsymbol{B}}_{\boldsymbol{pol}}$ is large and the flaring angle of the jet $\theta = \tan^{-1}(B_r/B_z)$ is nearly 0 (i.e. $B_z \gg B_r$**)**. Further away from the footpoint, the poloidal magnetic field is flared ($B_z \sim B_r$) but weak $|\bar{\boldsymbol{B}}_{\mathrm{pol}}| \to 0$, so the canonical vorticity becomes more axial $\bar{\boldsymbol{\Omega}}_{\mathbf{ipol}} \sim \bar{\omega}_{iz}\hat{\boldsymbol{z}}$ resulting in a collimated canonical flux tube Ψ_i. The location up the axis where the flux tube is fully collimated depends on where $\omega_{iz} \gg |\boldsymbol{B}_{\boldsymbol{pol}}|$ and occurs closer to the starting footpoint at larger flow momentum $|\bar{\boldsymbol{u}}_{\boldsymbol{e}}|$ and $\bar{\lambda}$ values (Figures 3.6(a)-(b)).

Because the current density is self-similar along the magnetic flux tube, ion canonical vorticity streamlines that depart magnetic flux surfaces travel through different regions within the current profile. Different strengths of current density change the strength of the ion flow velocities (equation 3.6) and therefore change the contribution of ω_{ir} and ω_{iz} to the canonical vorticity $\boldsymbol{\Omega}_{\boldsymbol{i}}$. For the uniform current case, $|\bar{\omega}_{ir}| \ll |\bar{\omega}_{iz}|$ everywhere and can be ignored. For the diffuse core current, however, there is a region near the exterior of the jet where $\bar{\omega}_{ir} < 0$ and $|\bar{\omega}_{ir}| \gtrsim |\bar{\omega}_{iz}|$ at low $\bar{\lambda}$ values (Figure 3.6(c)). The axial flow vorticity $\bar{\omega}_{iz}(r)$ is also strongly peaked on axis. These two conditions influence where ion canonical flux surfaces depart the magnetic flux surface and where complete collimation occurs. Therefore, the ion canonical flux tube collimates closer to the base of the jet in the diffuse core current case (Figure 3.6(c)) than for the uniform current case (Figure 3.6(a)) at low $\bar{\lambda}$. This is true for both sub- and super-Alfvénic electron flow velocities (Figure 3.5(a), (e) and Figure 3.5(c), (g) respectively). At high $\bar{\lambda}$ however, $\bar{\omega}_{iz}$ dominates the effect of $\bar{\omega}_{ir}$ so the collimation location and radii is not so sensitive to the current profile (Figure 3.5(b) versus 3.5(d); Figure 3.5(f) versus 3.5(e); Figure 3.6(b) versus 3.6(d)).

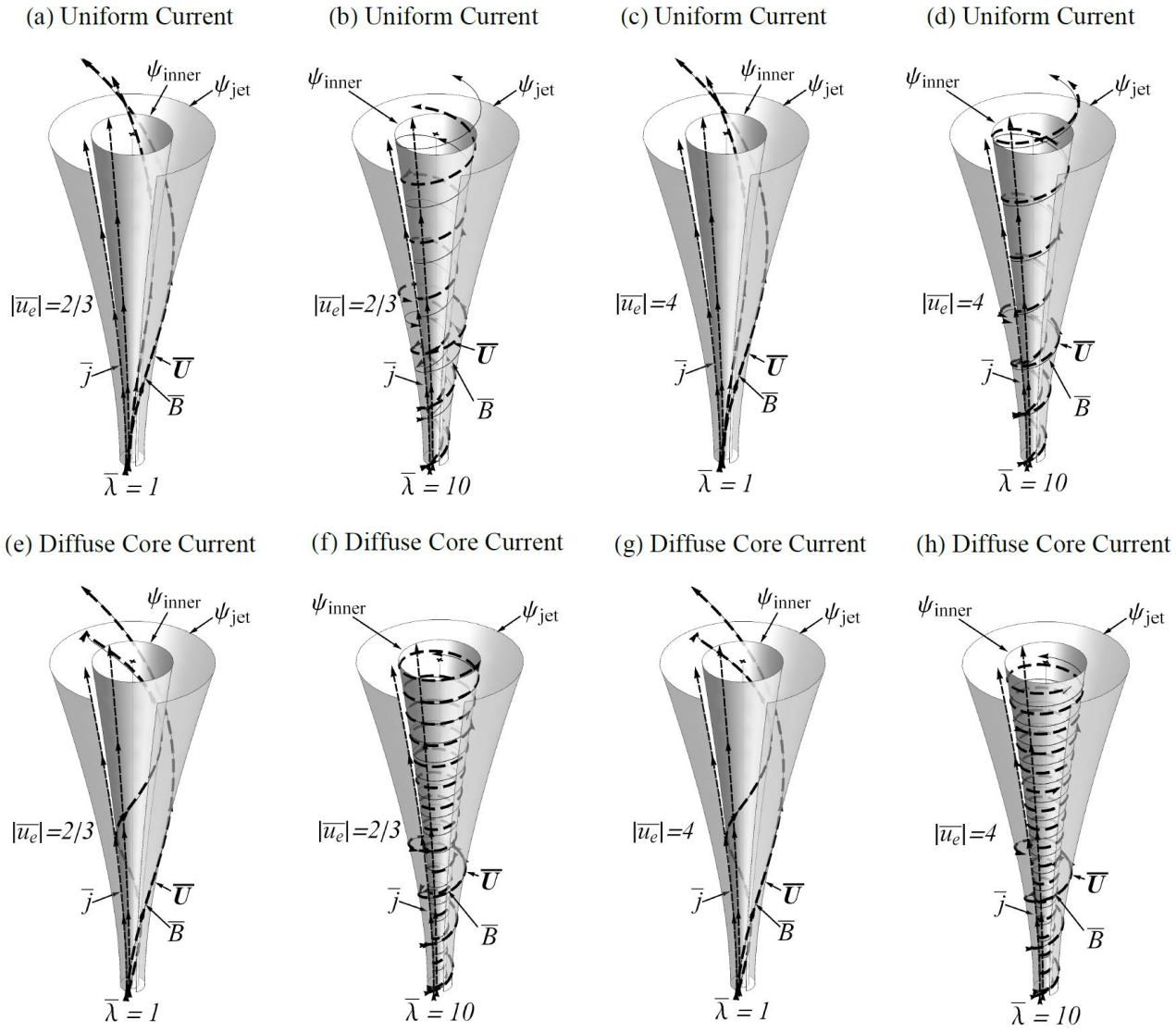


Figure 3.4. Flared magnetic flux tubes with uniform and diffuse core current profiles for sub-Alfvénic ($|\bar{\boldsymbol{u}}_e| = 2/3$) and super-Alfvénic ($|\bar{\boldsymbol{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. Thin, solid lines are magnetic field streamlines ($\bar{\boldsymbol{B}}$), thin dashed lines are current density streamlines ($\bar{j}$), and bold dashed lines are bulk flow velocity streamlines ($\bar{\boldsymbol{U}}$). Arrows on the streamline indicate the direction of the vector field. Streamlines on both the outer magnetic flux surface (ψ_{jet}) and an inner magnetic flux surface (ψ_{inner}) are shown; however, some streamlines on the outer magnetic flux surface have been truncated for visual clarity. Figure 5 shows the same systems from a canonical flux tube point-of-view.

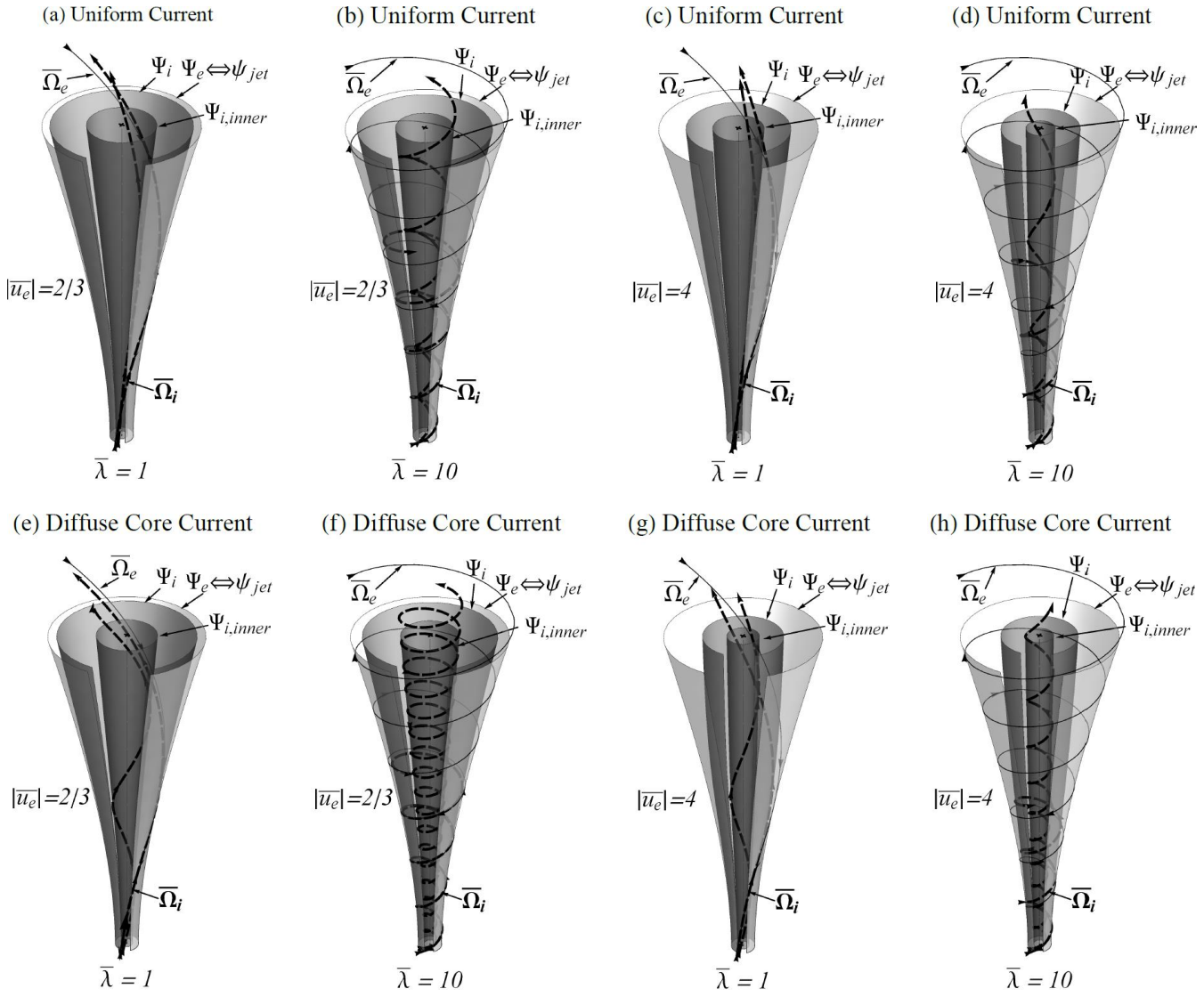


Figure 3.5. Canonical vorticity flux tubes with uniform and diffuse core current profiles for sub-Alfvénic ($|\overline{\boldsymbol{u}}_e| = 2/3$) and super-Alfvénic ($|\overline{\boldsymbol{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. These canonical flux tubes are equivalent to the corresponding magnetic flux tubes of Fig. 4. Ion canonical vorticity streamlines ($\overline{\boldsymbol{\Omega}}_{\boldsymbol{i}}$) are drawn as bold, dashed lines along two ion canonical vorticity flux tubes (Ψ_i and $\Psi_{i,inner}$), while electron canonical vorticity ($\overline{\boldsymbol{\Omega}}_{\boldsymbol{e}}$) streamlines (geometrically equivalent to magnetic field streamlines), are depicted as a thin, solid line along an electron canonical flux tube ($\Psi_e \Leftrightarrow \psi_{jet}$, light gray tube). Some streamlines on the outer ion canonical vorticity flux tube (Ψ_i, dark gray tubes) have been truncated for visual clarity. As current is increased, ion canonical flux tubes depart from magnetic flux tube surfaces and become collimated.

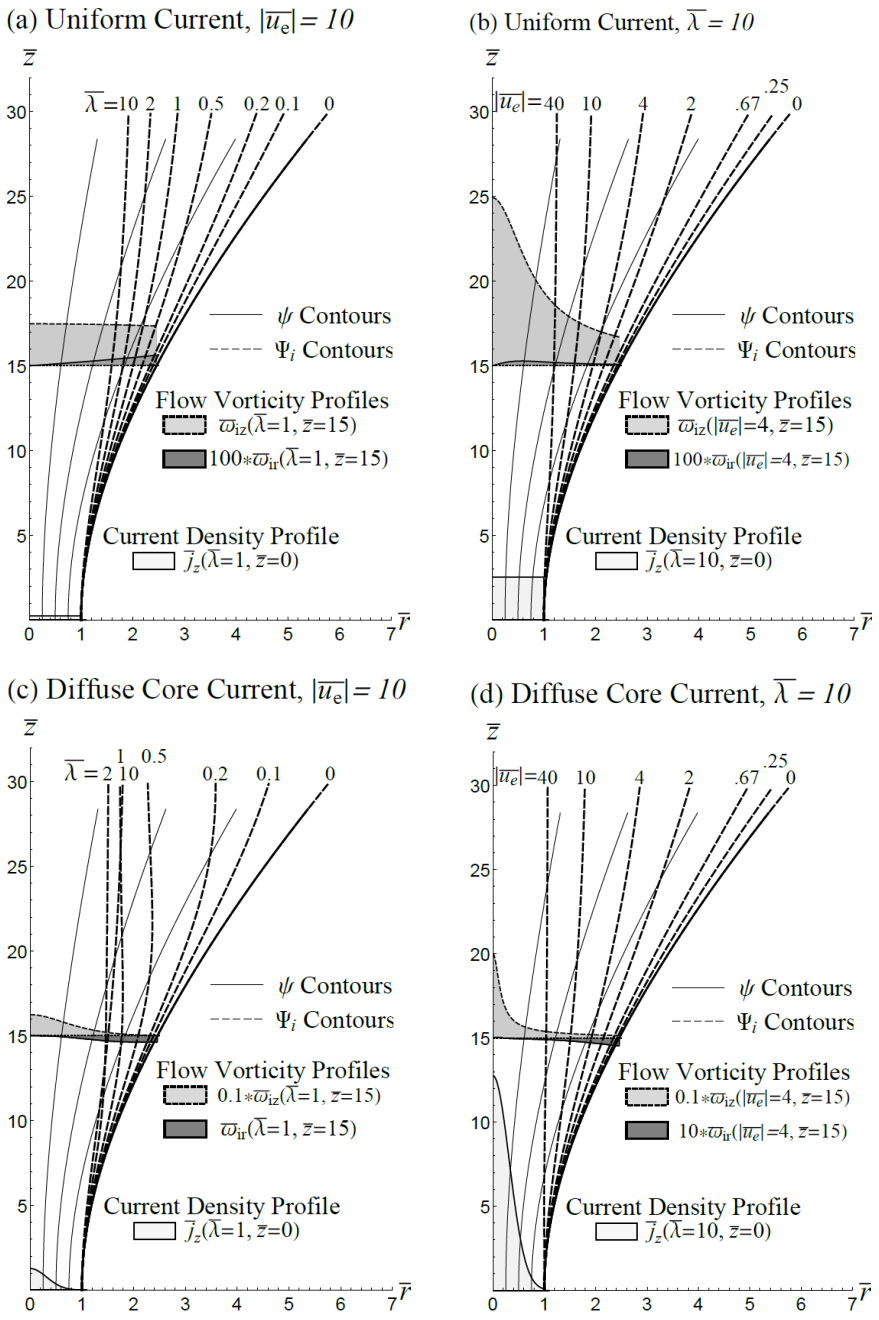


Figure 3.6. Dependence of ion canonical vorticity flux surface Ψ_i (dashed lines) on total jet current $\bar{\lambda}$ (panels a, c) and on electron fluid velocity $|\overline{\boldsymbol{u}}_e|$ (panels b, d) for uniform (panels a, b) and diffuse core (panels c, d) currents. For both current profiles, full collimation occurs closer to the base of the flux tube with increasing electron velocity $|\overline{\boldsymbol{u}}_e|$. A prominent bulge in the canonical flux tube occurs at low $\bar{\lambda}$~0.5 and high $|\overline{\boldsymbol{u}}_e| = 10$ for a diffuse core current profile (panel c). Plots of current profile $\bar{j}_z(\bar{r})$ are shown at the base of the jet (lightest gray) while representative poloidal flow vorticity profiles $\bar{\omega}_{ir}(\bar{r})$ (dark gray) and $\bar{\omega}_{iz}(\bar{r})$ (light gray) are drawn at $\bar{z} = 15$. Contours of constant magnetic flux ψ (solid lines) do not change with $\bar{\lambda}$ nor $|\overline{\boldsymbol{u}}_e|$ and are presented to reveal passage of the canonical flux surface through different regions of the current profile.

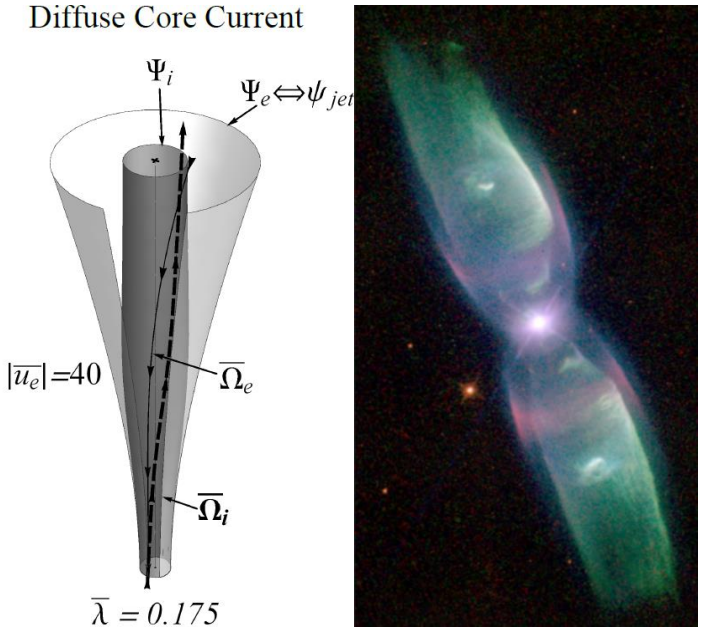


Figure 3.7. (Left) Bulged ion canonical vorticity flux tube Ψ_i (dark gray) for a diffuse core current jet with low current ($\bar{\lambda} = 0.175$) and a large electron velocity ($|\overline{\boldsymbol{u}}_e| = 40$). Equation (5) shows that density gradients can be normal to the dark gray canonical Ψ_i surfaces rather than the light gray magnetic flux surfaces ψ suggesting that ion line emission reveals canonical flux tubes as opposed to magnetic flux tubes. **(Right)** M2-9 Butterfly nebula (NASA HST STIS/CCD – MIRVIS), a bipolar planetary nebula with distinctive bulging that resembles the ion canonical flux tube in the left panel. Transitions in the geometry of the jet (i.e. flaring to collimation) may therefore also reflect changes in the relative strength of the fluid flow vorticity flux to the magnetic flux. Image credit: Bruce Balick (University of Washington), Vincent Icke (Leiden University), Garret Mellema (Stockholm University), and NASA

When $\bar{\omega}_{ir} < 0$ and $|\bar{\omega}_{ir}| > |\bar{B}_r|$, ion canonical flux surfaces can be pulled inward ($-\hat{\boldsymbol{r}}$ direction) resulting in a bulged canonical flux tube that narrows before collimating at a constant final collimation radius a_{Ψ_i} (Figure 3.7). This occurs for the diffuse core current jet at low $\bar{\lambda}$ and large $|\overline{\boldsymbol{u}}_e|$ values, e.g. Ψ_i contour in Figure 3.6(c) with $\bar{\lambda} = 0.5$ and $|\overline{\boldsymbol{u}}_e| = 10$ where the bulge is visible at $\bar{z}$~20 and $\bar{r}$~2.4. As $\bar{\lambda}$ is increased and $\bar{\omega}_{iz}$ begins to dominate, any bulging or narrowing becomes less prominent and the final collimation radius $a_{\Psi_i}(\bar{l})$ actually becomes slightly larger again (Ψ_i contours in Figure 3.6(c) with $\bar{\lambda} = 10$ with $a_{\Psi_i}(\bar{l})$~1.8 versus $\bar{\lambda} = 2$ with $a_{\Psi_i}(\bar{l})$~1.4). For both current profiles, the final collimation radius appears to approach a constant value with increasing $\bar{\lambda}$ and $|\overline{\boldsymbol{u}}_e|$ values, i.e. for a fixed $|\overline{\boldsymbol{u}}_e|$ the final collimation radius approaches a constant and will not change further with increasing $\bar{\lambda}$; and for a fixed $\bar{\lambda}$, the final collimation radius approaches a different constant with increasing $|\overline{\boldsymbol{u}}_e|$. For the case of the uniform current profile,

the approach toward smaller radii is monotonic with increasing $\bar{\lambda}$ and $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$. For the diffuse core current, however, the approach is monotonic with $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ (Figure 3.6(d)), but not with $\bar{\lambda}$ (Figure 3.6(c)) where a minimum collimation radius $a_{\Psi_i}(\bar{l})\sim 1.4$ is observed at $\lambda\sim 2$. Since (in reality) ions should be tied to canonical vorticity flux surfaces (equations 2.7, 2.9, 2.12), any light emission from ions would be signatures of canonical vorticity flux tubes rather than magnetic flux surfaces, with any associated bulges and collimation dependent on the current and flow profiles (and because of the massless electron assumption here, electron emission would be signatures of magnetic flux surfaces).

3.3.2. Diffuse-core-with-skin and skin-only current profiles

Figure 3.8 presents the morphology of a flared plasma jet with diffuse-core-with-skin and skin-only current profiles from the MHD perspective of current-carrying magnetic flux tubes at low and high $\bar{\lambda}$ with a bulk velocity $\overline{\boldsymbol{U}}$ also effected by the choice of $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$. Again, as $\bar{\lambda}$ is increased, the magnetic field becomes helical in regions that enclose a net current. For the diffuse-core-with-skin current profile, this occurs throughout the jet with a shear due to gradients in current density profile $j_z(r)$. However, for the case of the skin current, the magnetic field in the core of the jet ($\eta < 0.6$) remains untwisted because of the absence of current in this region. This results in significant shear in the skin region because B_θ must increase over a shorter distance to match the final value at the boundary to satisfy Ampère's law. Bulk flow separation from magnetic field streamlines is again apparent in regions of finite current density, with lower electron velocities resulting in increasing pitch angle. Because there is no toroidal current ($j_\theta = 0$), magnetic flux surfaces remain unchanged as the current is increased. From a canonical flux tube point-of-view (Figure 3.9), the shape of canonical flux surfaces depends on the species flow velocity as well as on jet current and current profile (Figure 3.10). Figures 3.9(a)-(d) show the equivalent systems to Figures 3.8(a)-(b) (diffuse-core-with-skin profile) for the same electron flow velocities used in section 3.3.1, while Figures 3.9(e)-(h) show equivalent systems to Figures 3.8(c)-(d) (skin current profile). Again, the electron canonical vorticity flux tubes are geometrically coincident to the magnetic flux tubes $\Psi_e \Leftrightarrow \psi_{jet}$ because of our massless electron assumption, while the ion canonical vorticity flux surfaces depart magnetic flux surfaces $\Psi_i \nLeftrightarrow \psi_{\mathrm{jet}}$ (in regions of finite $\boldsymbol{\omega}_{\mathbf{ipol}}$) to become increasingly collimated further up away from the footpoint. For the diffuse-core-with-skin current

profile, collimation is complete in regions closer to the footpoint at larger currents $\bar{\lambda}$ and electron flow velocities $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$, whereas for the skin current profile, collimation is never complete due to the absence of current in the core region of the jet (Figure 10).

Similar to section 3.3.1, for both current profiles $\bar{\omega}_{iz} \geq 0$ everywhere since the total current increases monotonically with radius. For the diffuse-core-with-skin current, the radial profile of $\bar{\omega}_{iz}(\bar{r})$ is peaked on axis with a secondary local maximum at $\eta = 0.8$ (i.e. peak of the skin current). For identical jet parameters $\bar{\lambda}$ and $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$, the magnitude of $\bar{\omega}_{iz}$ in the core region falls between that of the uniform and diffuse core current jets of section 3.3.1. However, due to the contribution of a peaked skin current, the magnitude of ω_{iz} in the skin region exceeds that of the uniform and diffuse core current profiles for similar jet parameters. As a result, the narrowing and bulging effect observed for the diffuse-core current jet at low $\bar{\lambda}$ and high $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ (Figure 3.6(c)) is less pronounced even though a finite, negative $\bar{\omega}_{ir}$ is observed near the jet boundary (Figure 3.10(a)). A transition within the skin region of $\bar{\omega}_{ir}$ from negative to positive (moving in the inward radial direction) further restricts any narrowing of the canonical flux tube. Consequently, for identical jet parameters, collimation of the diffuse-core-with-skin current jet occurs closer to the footpoint than for uniform current, but farther away than for the purely diffuse core current jet. The approach toward tighter radii is monotonic with increasing $\bar{\lambda}$ and $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ (Figures 3.10(a)-(b)) as was the case for the uniform current profile (Figures 3.6(a)-(b)).

For the skin current profile, the absence of current in the core region of the jet prevents collimation of the ion canonical flux tube. With no poloidal current in this region, magnetic fields are untwisted as are the species velocity fields; consequently, $\boldsymbol{\omega}_{\mathbf{ipol}} = 0$ and the ion canonical flux surfaces $\Psi_{i,\mathrm{inner}}$ are equivalent to magnetic flux surfaces ψ_{inner}. In the skin region however, a large radial current gradient results in a strongly peaked $\bar{\omega}_{iz}(\bar{r})$ profile that works to collimate canonical flux surfaces Ψ_i. The presence of a negative $\bar{\omega}_{ir}$ at the jet boundary further helps to separate canonical flux surfaces, but any narrowing of the flux tube is again constrained by a transition (in the inward radial direction) of $\bar{\omega}_{ir}(r)$ to positive values (Figure 10(c)). As $\bar{\lambda}$ and $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ are increased, canonical flux surfaces approach the current free region of the jet but can go no further (Figures 10(c)-(d)). Because the current is self-similar along the magnetic flux tube, the current free region is flared and therefore the ion canonical flux tubes can never fully collimate but must follow the less-pronounced flaring of the current free region. The shell between the ion

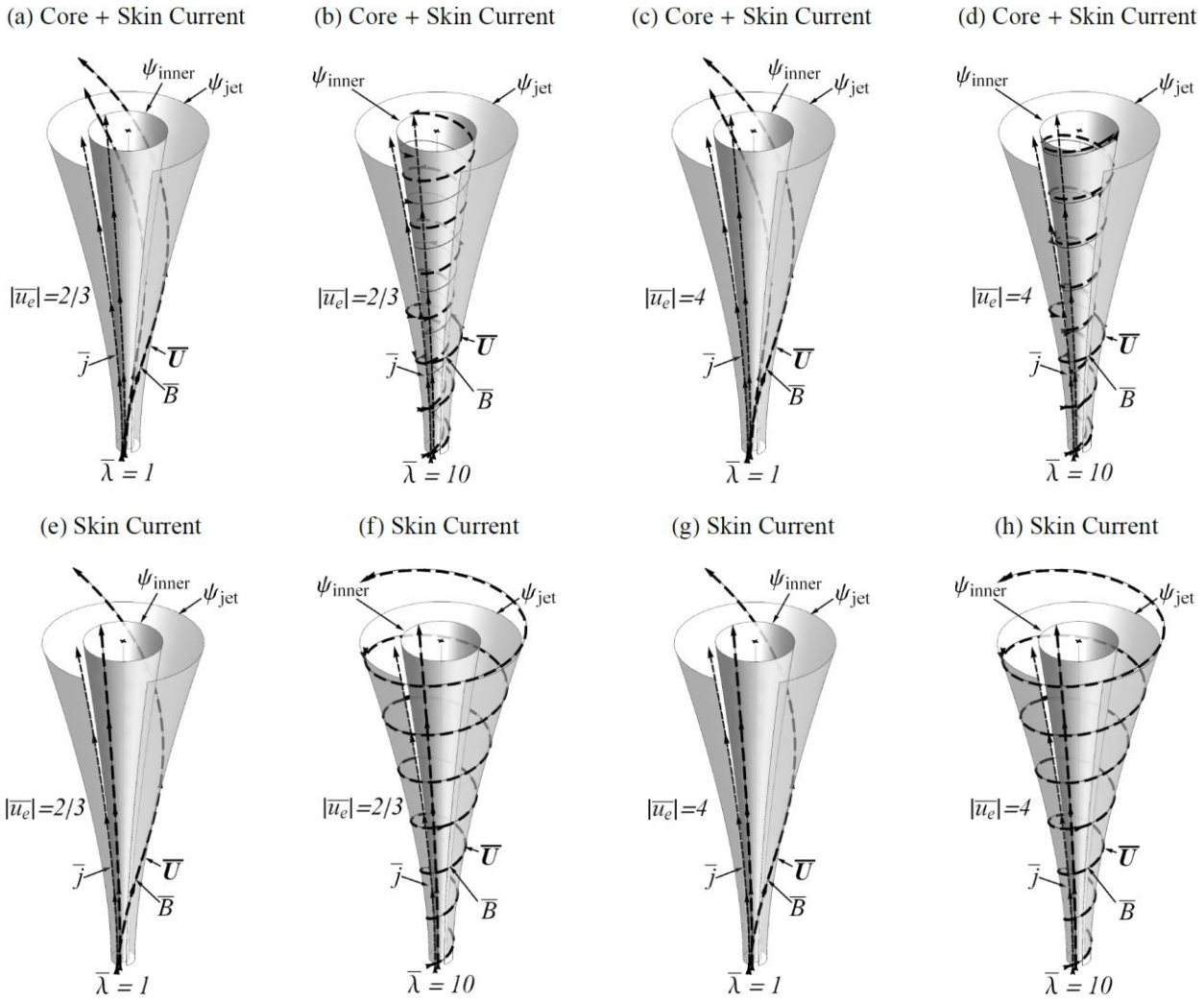


Figure 3.8. Flared magnetic flux tubes with diffuse-core-with-skin and skin-only current profiles for sub-Alfvénic ($|\overline{\boldsymbol{u}}_e| = 2/3$) and super-Alfvénic ($|\overline{\boldsymbol{u}}_e| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. Thin solid lines are magnetic field streamlines ($\overline{\boldsymbol{B}}$), thin dashed lines are current density streamlines ($\bar{j}$), and bold dashed lines are bulk flow velocity streamlines ($\overline{\boldsymbol{U}}$). Arrows on the streamline indicate the direction of the vector field. Streamlines on both the outer magnetic flux surface (ψ_{jet}) and an inner magnetic flux surface (ψ_{inner}) are shown; however, some streamlines on the outer magnetic flux surface have been truncated for visual clarity. Figure 5 shows the same systems from a canonical flux tube point-of-view.

canonical flux tube (e.g. ion line emission) and the magnetic flux tube (e.g. electron synchrotron emission) then represents the skin current density shell.

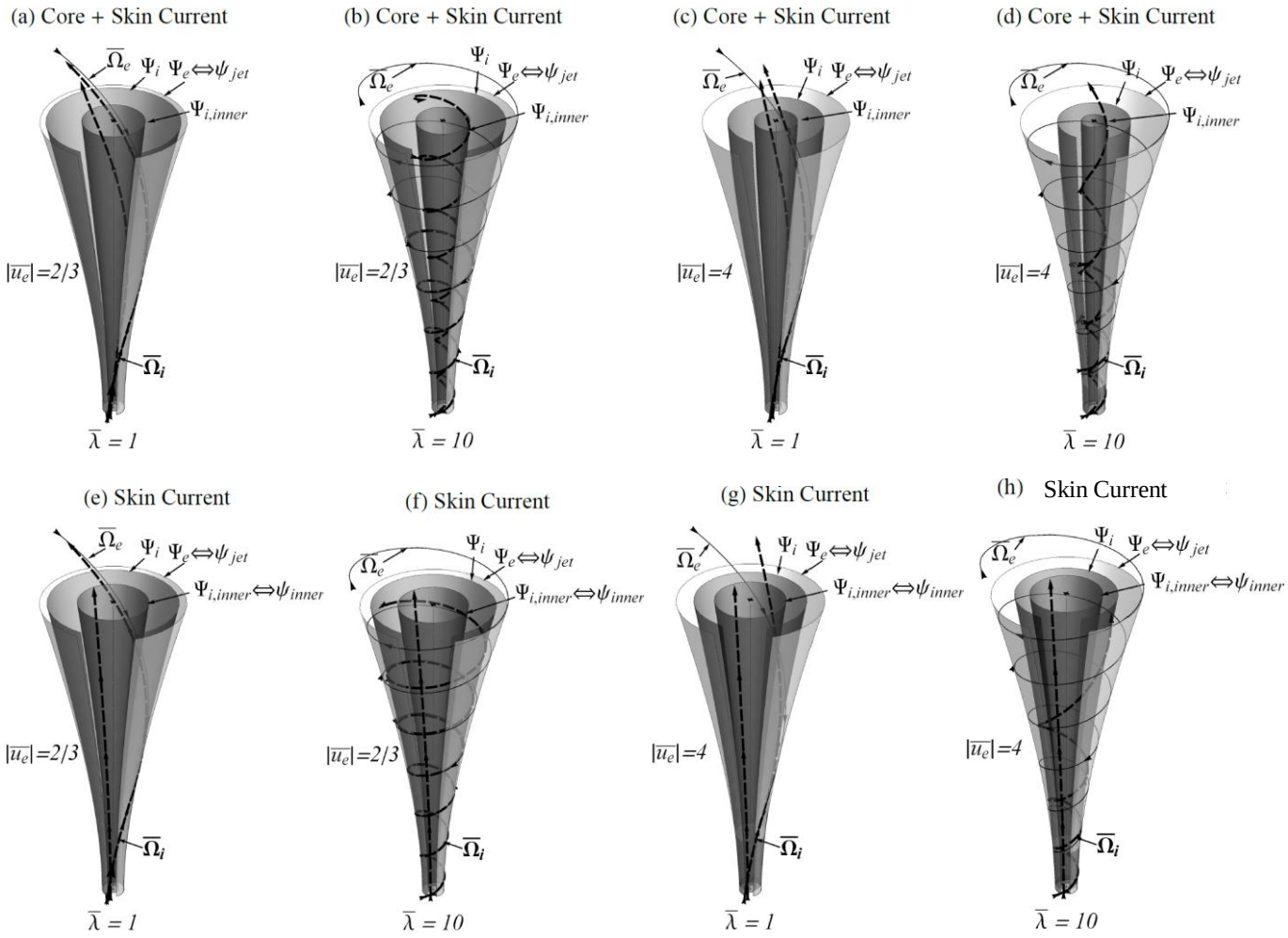


Figure 3.9. Canonical vorticity flux tubes with diffuse-core-with-skin and skin-only current profiles for sub-Alfvénic ($|\bar{\boldsymbol{u}}_{\boldsymbol{e}}| = 2/3$) and super-Alfvénic ($|\bar{\boldsymbol{u}}_{\boldsymbol{e}}| = 4$) electron flow velocities at low $\bar{\lambda} = 1$ and high $\bar{\lambda} = 10$. These canonical flux tubes are equivalent to the corresponding magnetic flux tubes of Figure 8. Ion canonical vorticity streamlines ($\bar{\boldsymbol{\Omega}}_{\boldsymbol{i}}$) are drawn as bold, dashed lines along two ion canonical vorticity flux tubes (Ψ_i and $\Psi_{i,inner}$), while electron canonical momentum ($\bar{\boldsymbol{\Omega}}_{\boldsymbol{e}}$) streamlines (geometrically equivalent to magnetic field streamlines), are depicted as a thin, solid line along an electron canonical flux tube ($\Psi_e \Leftrightarrow \psi_{jet}$, light gray tube). Some streamlines on the outer ion canonical vorticity flux tube (Ψ_i, dark gray tubes) have been truncated for visual clarity. As the current is increased, ion canonical flux tubes depart from magnetic flux tube surfaces and begin to collimate. For the skin current (e-h), collimation never completes due to the absence of poloidal flow vorticity flux in the core of the jet but the flaring is less pronounced than the flare of the magnetic flux tube.

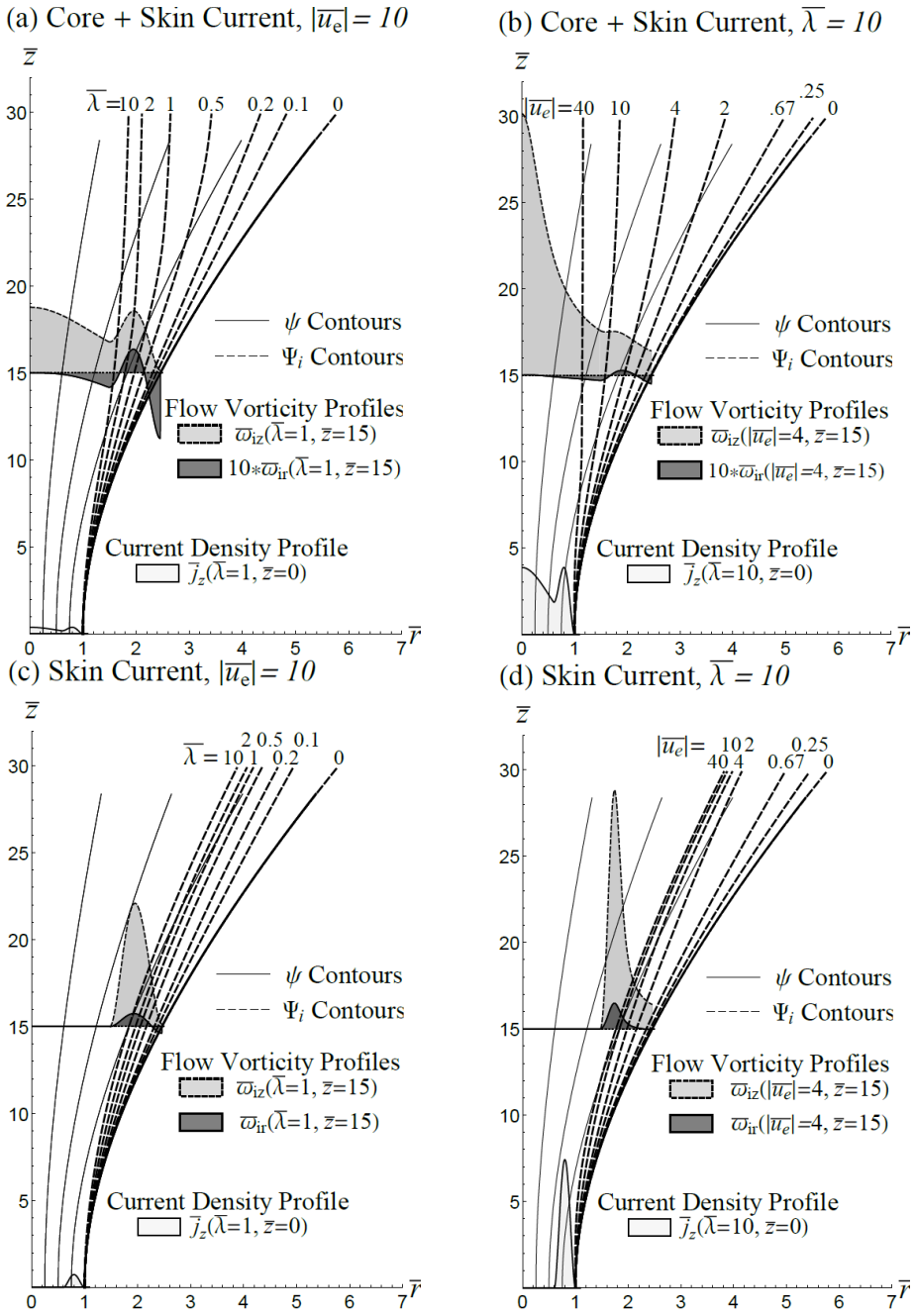


Figure 3.10. Dependence of ion canonical vorticity flux surface Ψ_i on total jet current $\bar{\lambda}$ (panels a, c) and on electron fluid velocity $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ (panels b, d) for diffuse-core-with-skin (panels a, b) and skin-only (panels c, d) currents. For both current profiles, canonical flux surfaces separate closer to the base of the jet with increasing jet current $\bar{\lambda}$ and electron velocity $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$; however, for the skin-only current, collimation does not occur due to the absence of poloidal flow vorticity flux in the core of the jet. Plots of the current profile $\bar{j}_z(\bar{r})$ are shown at the base of the jet (lightest gray) while representative poloidal flow vorticity profiles $\bar{\omega}_{ir}(\bar{r})$ (dark gray) and $\bar{\omega}_{iz}(\bar{r})$ (light gray) are drawn at $\bar{z} = 15$. Contours of constant magnetic flux ψ (solid lines) do not change with $\bar{\lambda}$ nor $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}|$ and are presented to reveal passage of the canonical flux surface through different regions of the current profile.

3.4. Kinematic evolution of relative canonical helicity

Figure 3.11 shows the kinematic evolution of the normalized (weighted) ion relative canonical helicity components (magnetic helicity $\overline{\mathcal{K}}$, cross-helicity $\overline{\mathcal{X}}_i$, kinetic helicity $\overline{\mathcal{H}}_i$ components of the canonical helicity $\overline{K}_{irel}$ in equation (24)) for all cases considered in Section 3.3 as $\bar{\lambda}$ is increased from 10^{-4} to 40. Integrations are performed using a trapezoidal method with a precision of $\pm 1\%$. The normalized helicity values have been transformed using a logarithmic modulus transformation $K' = \text{sign}(K)\log_{10}(|K| + 1)$ that preserves any zero crossings. The electron canonical helicity, although not shown explicitly, is equivalent to the weighted magnetic helicity component because of our massless electron assumption. These plots demonstrate the influence of fluid momentum and current profile on canonical helicity, while revealing distinct regions in $\bar{\lambda}$ space where different helicity components dominate.

For the current profiles and electron velocities considered in section 3.3, cross and magnetic helicity dominate the total helicity content of the jet (Figure 3.11). At low $\bar{\lambda}$, the cross-helicity dominates because ions flow roughly along the magnetic field lines with a toroidal vorticity parallel to the toroidal magnetic vector potential. As $\bar{\lambda}$ increases, magnetic helicity increases sufficiently rapidly to dominate cross-helicity above a threshold $\bar{\lambda}$. The threshold where this interchange occurs is around $0.001 < \bar{\lambda} < 0.01$. For cases with larger flow velocities ($|\overline{\boldsymbol{u}}_e| = 4$ in Figure 11(d), (f), (h)), as $\bar{\lambda}$ is increased above another threshold around $\bar{\lambda}\sim 2$, cross-helicity once again dominates magnetic helicity due primarily to a large poloidal flow vorticity and poloidal magnetic vector potential.

At larger electron velocities, the magnitude of cross and flow helicity increases while the overall shape of their curves is maintained. By increasing the electron velocity well above $|\overline{\boldsymbol{u}}_e| = 4$, (i.e., approaching neutral fluid limit), scenarios emerge in which the ion flow helicity can dominate (followed by cross-helicity) over large regions of $\bar{\lambda}$ space. This is demonstrated in Figure 3.12, where the normalized electron velocity is set to $|\overline{\boldsymbol{u}}_e| = 40$ for a uniform current jet. In this case, ion kinetic helicity grows with magnetic helicity at low $\bar{\lambda}$ and becomes the dominant helicity component between $0.2 \lesssim \bar{\lambda} \lesssim 10$. In regions outside of this range the cross-helicity dominates all other helicity components.

In all cases, a reduction in the growth of ion flow helicity $\overline{\mathcal{H}}_i$ is seen at high $\bar{\lambda}$. This can be explained by noting that, at all locations where the current density is large, ions must

compensate for any toroidal electron flow and increase their pitch angle (i.e. untwist). This unravelling of ion flow tubes is illustrated in Figure 3.13 for the case of a uniform current jet with sub-Alfvénic electron velocities. Because ion flows only unwind in regions of finite current density, the impact on the ion flow helicity $\bar{\mathcal{H}}_i$ is most pronounced for jets in which the current density is more uniformly distributed throughout the current-carrying region. In fact, for the case of the uniform current and diffuse-core-with-skin current profiles, the ion flow helicity is observed to decrease for $\bar{\lambda} \gtrsim 10$ whereas for the diffuse-core current and skin current profiles, the decrease in growth rate is less severe.

Another interesting feature is the negative dip in ion flow helicity for the case of the diffuse core current (Figure 11 (c)-(d)). To explain this, the ion relative flow helicity can be expanded as

$$\bar{\mathcal{H}}_{irel} = \iiint \bar{u}_{ir}\left(\bar{\omega}_{ir} + \bar{\omega}_{ir,ref}\right) + \left(\bar{u}_{i\theta} - \bar{u}_{i\theta,ref}\right)\bar{\omega}_{i\theta} + \bar{u}_{iz}\left(\bar{\omega}_{iz} + \bar{\omega}_{iz,ref}\right)d\bar{V}. \tag{3.17}$$

The reference fields are obtained from equation (3.15) and are generally opposite in sign to the actual fields as a result of the boundary conditions enforced to ensure gauge independence. For the diffuse core current, the first term in the integrand is small and negative over the $\bar{\lambda}$ range since $\bar{\omega}_{ir}$ is generally negative (Figure 6 (c)-(d)) because of the shape of the current profile and flaring of the jet (u_{ir} is always positive or zero). The reference field $\bar{\omega}_{ir,ref}$ in this case does not fully cancel $\bar{\omega}_{ir}$ when integrated over the jet volume. Because the axial flow velocity $\bar{u}_{iz}$ and vorticity $\bar{\omega}_{iz}$ are always positive, the third term in the integrand is always positive (again the reference field does not fully cancel the actual field upon integration). This term grows with increasing $\bar{\lambda}$ as the magnitude of $\bar{u}_{iz}$ and $\bar{\omega}_{iz}$ increases within the jet. The second term in the integrand varies mostly due to the complicated dependence of $\bar{\omega}_{i\theta}$ on the current profile and flaring angle of the jet. Using equation (3.6) to expand the toroidal vorticity in terms of the electron velocity and current density

$$\bar{\omega}_{i\theta} = \frac{\partial}{\partial \bar{z}}\left(\bar{J}_r + \bar{u}_{er}\right) - \frac{\partial}{\partial \bar{r}}\left(\bar{J}_z + \bar{u}_{ez}\right) \tag{3.18}$$

it can be seen that at zero jet current ($\partial \bar{J}_r/\partial \bar{z} = \partial \bar{J}_z/\partial \bar{r} = 0$), $\bar{\omega}_{i\theta}$ is finite and positive because electrons flow with a constant velocity along flared magnetic field lines. The radial profile $\bar{\omega}_{i\theta}(\bar{r})$ peaks at the wall of the jet; however, as the current is increased, the toroidal ion flow vorticity

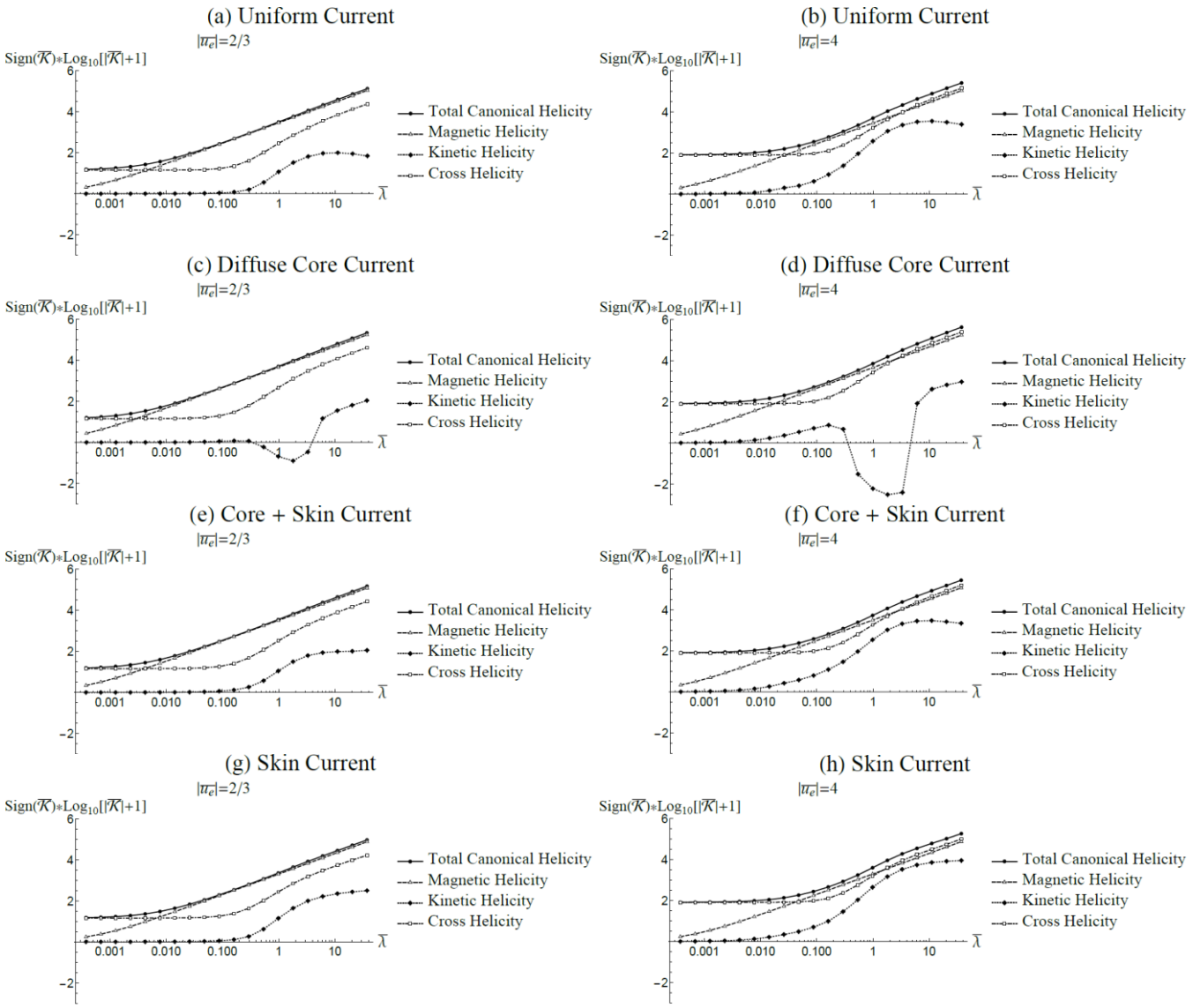


Figure 3.11. Kinematic evolution of the weighted normalized ion relative canonical helicity components for select current profiles and electron flow velocities. For all the cases considered in Section 4 cross and magnetic helicity components dominate the total canonical helicity content of the jet. Ion kinetic helicity increases at moderate values of $\bar{\lambda}$ but levels off or even decreases as the ion flows unwind at high $\bar{\lambda}$ (see the text, Figure 13). For the diffuse core current profile (panels c, d), a negative dip in the ion flow helicity is observed at moderate $\bar{\lambda}$ as a consequence of a changing $\bar{\omega}_{i\theta}(r)$ profile.

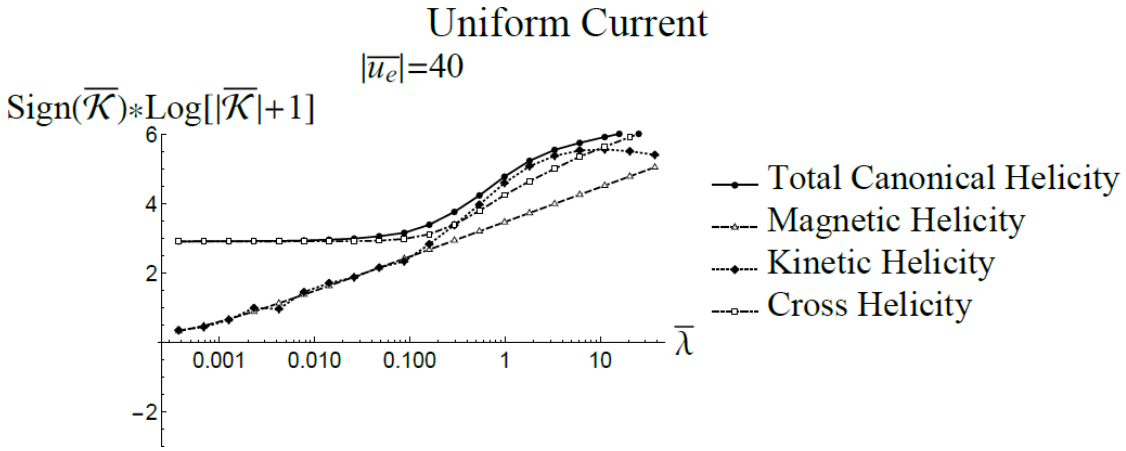


Figure 3.12. Kinematic evolution of the normalized ion relative canonical helicity components for a uniform current profile with a normalized electron flow velocity $|\overline{\boldsymbol{u}}_{\boldsymbol{e}}| = 40$. In this case, the ion kinetic helicity dominates at moderate $0.2 \lesssim \bar{\lambda} \lesssim 10$, with cross-helicity dominating at larger $\bar{\lambda}$.

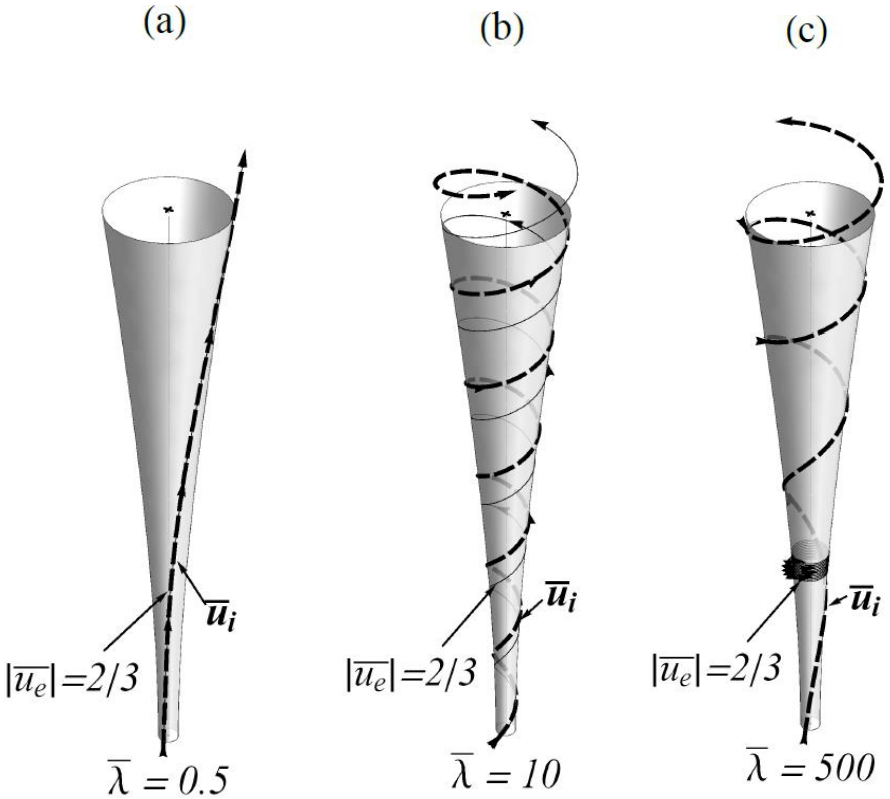


Figure 3.13. Ion flow tubes for the case of a uniform current profile with increasing jet current $\bar{\lambda}$. Ion flow streamlines (dashed line) twist as $\bar{\lambda}$ is increased (panel a to b) then unwind at even larger $\bar{\lambda}$ (panel b to c) to compensate for increasing electron twist (solid line) in carrying the poloidal current. This behavior explains the reduction in ion flow helicity observed at large $\bar{\lambda}$ values for current profiles that are more distributed throughout the cross-section of the jet (Figure 3.11).

decreases in the outer region as a consequence of the negative gradient in the toroidal magnetic field (i.e. increasing $|\bar{u}_{ez}|$ with $\bar{r}$) while increasing in the core due to the negative radial gradient in current density and positive gradient in the toroidal magnetic field (i.e. decreasing $|\bar{u}_{ez}|$ with $\bar{r}$). Meanwhile, the first term on the right of equation (3.18) remains negative and small compared to the second because of the shallow flaring of the jet. As the current is increased, the volume where $\omega_{i\theta} < 0$ (and $\bar{u}_{i\theta}$ is large) begins to exceed that where $\omega_{i\theta} > 0$ and term 2 in equation (3.17) becomes negative. This term grows faster in magnitude than term three and causes the ion flow helicity to dip negative around $\bar{\lambda}\sim 0.3$. Increasing the jet current beyond $\bar{\lambda}\sim 1.5$ and the toroidal ion flow vorticity becomes increasingly peaked in the core of the jet, causing the integral term to increase. This together with the growth of the third term in equation (3.17) is responsible for the negative-to-positive transition observed in the relative ion flow helicity at $\bar{\lambda}\sim 3$.

3.5. Summary

This chapter has presented a more fundamental framework for examining magnetized flows in a jet geometry by examining the geometry and topology of canonical fields in a flared current-carrying magnetic flux tube. In essence, this single framework provides a transformation from the familiar equations of motion to a generalized Maxwell equation field formulation [48], to demonstrate that the canonical flux tube geometry point-of-view is valid in regimes beyond the fluid regime, and in particular, can be applied to kinetic and relativistic regimes. The self-consistent dynamics at kinetic regimes is left to future work, while the stationary reduced two-fluid cases are considered here. By varying the enthalpy boundary conditions (i.e. current profile and flow velocity profile), we reveal the dependence of canonical fields on the underlying magnetic and flow components and show that canonical flux tubes can become long and collimated even when the magnetic field is flared. For the current profiles considered, ion canonical flux tubes collimate as a result of positive flow vorticity flux. The final radius and location at which collimation occurs was shown to depend on the total jet current $\bar{\lambda}$, current profile $\bar{J}_z(r)$, and flow momentum $|\bar{\boldsymbol{u}}_{\boldsymbol{e}}|$. Taken with the fact that the curl of equation 2.7 in the absence of dissipation generalizes the frozen-in magnetic flux condition to the more fundamental frozen-in canonical vorticity flux, our results suggest that collimated astrophysical jets in nature may trace canonical vorticity flux tubes as opposed to magnetic flux tubes. Transitions in the shape of such jets (i.e. flaring, bulging, and collimation) could therefore reflect variation in the relative strength of the

magnetic and fluid vorticity fluxes. The tendency of canonical flux tubes to collimate agrees with observations that some astrophysical jets become increasingly collimated further away from their origin [62].

By accounting for finite particle momentum, the canonical viewpoint preserves non-ideal effects that are lost in ideal MHD, while retaining intuition of the changing geometry of flux tubes. Concepts such as magnetic helicity conservation, which govern the evolution of magnetic flux tubes into other magnetic flux tubes, can therefore be generalized to canonical helicity conservation, which predicts the topological evolution of both magnetic and flow tubes. We calculate the gauge-invariant relative canonical helicity content of the jet and reveal the sensitivity of various (magnetic, cross, and kinetic) helicity components to the shape and magnitude of the current profile and electron flow velocity. At large jet currents, we observe a decrease in the ion kinetic helicity as ion flow fields unwind to compensate for increasing magnetic twist. We hypothesize that conversion between helicities at the appropriate scales [50, 60] could convert helical magnetic twist into helical shear flows, which could be sufficiently strong to stabilize against current driven instabilities. Testing this hypothesis would require solving for the full dynamics of the helicity transport equation and canonical equation of motion, and verification in a laboratory astrophysical jet experiment. The design, construction, and initial results of such an experiment are the subject of the remaining chapters in this thesis.

Chapter 4
THE MOCHI.LABJET EXPERIMENT

4.1. Introduction

The MoCHI.labjet experiment is a triple electrode planar plasma gun experiment constructed to study the dynamics of magnetically driven jets with strong helical shear flows. Of primary interest is improving our understanding of the interaction between flows, magnetic fields, and jet stability from the point of view of canonical flux tubes and canonical helicity. The LabJet configuration aims to simulate an astrophysical jet in a laboratory by mimicking an accretion disk threaded by a poloidal magnetic field with concentric planar electrodes in front of a bias magnetic field coil. Azimuthally symmetric gas injection slits provide a continuous mass injection source and allow for free rotation of the jet footpoints due to $\vec{E} \times \vec{B}$ rotation. This design therefore approximates a shear azimuthal rotation profile of and astrophysical accretion disk system. To measure canonical flux tubes and helicity, diagnostic measurements should, in theory, be capable of resolving the ion inertial lengths. For a laboratory plasma jet with a length of 1m, a diameter of $\sim 10\ cm$, and an ion inertial length of $1\ cm$, approximately 80,000 measurement points would be required to resolve this scale length over the entire volume [63].

A spherical vacuum chamber with numerous access ports was thus commissioned to provide a high level of diagnostic and spectroscopic access. Furthermore, an experimental control infrastructure was developed and implemented to create reproducible shots with a high throughput digitization to take as many measurements as possible per shot. This chapter discusses the design, control, and diagnostics of the experiment and presents results from the first plasmas.

4.2. Laboratory astrophysics

Plasmas constitute the bulk of visible matter within the universe yet rarely do they form naturally within earth's atmosphere. Measurements of such astrophysical plasmas are thus intrinsically difficult to obtain due to the vast distances and long timescales involved in their evolution. Within our solar system, satellites have proven to be a useful tool for both indirect and in-situe measurements. Outside of our solar system however, measurements are intrinsically indirect and often static since the evolution of large astrophysical systems can take place over thousands to millions of years. By recreating astrophysically relevant plasmas within a laboratory, we can

provide insight into plasma dynamics through direct measurements and repeated testing of pertinent phenomena.

Historically, there are two approaches to simulating astrophysical jet dynamics in a laboratory: high energy density (HED) jets driven by lasers or magnetic pinches [26, 64] and medium energy density magnetohydrodynamic jets driven by plasma guns (Z-pinches [37], screw pinches [32]). The first approach attempts to replicate or approach actual astrophysical dimensionless numbers [64] without regard for the boundary conditions of the formation region. In such experiments, the effect of high Mach number, radiative flows on jet collimation are studied through a careful control of experimental parameters and sophisticated diagnostics developed for inertial fusion research. In laser driven experiments [65-68], purely hydrodynamic or radiation dominated jets are driven by kJ lasers at high densities and velocities up to $\sim 100\ km/s$. Such jets are short-lived however, with lifetimes of 1-10 ns and lengths of $\sim 1 mm$. An alternative HED approach is based on mega-Ampère currents flowing through conically arranged radial wire arrays [69] or radial foils [70]. The large currents ablate the material, which collimates on axis due to $\vec{J} \times \vec{B}$ forces and/or non-uniform axial pressure gradients. High density jets several mm to cm long then propagate at velocities of ~ 200 km/s with aspect ratios of $\sim 5{:}\ 1$. Mach number, plasma beta, and cooling parameters in such experiments approach astrophysical jet conditions (except for the extreme aspect ratios) while extended MHD effects [70] and radiative cooling appear to play an important role in the collimation. Such an approach is experimentally advantageous as it provides a means to study jet propagation dynamics at relevant dimensionless parameters without having to replicate the rotating boundary conditions of an accretion disk, which would need to rotate impractically fast in such a laboratory setup.

In nature however, the physical parameters and dimensionless numbers of astrophysical jet systems can vary over many orders of magnitude [32, 71, 72]. Common features of such systems suggest that boundary conditions likely play a more important role than exact dimensionless numbers in jet dynamics and evolution. Medium energy density experiments therefore seek to model the global behavior of astrophysical jets by setting appropriate boundary conditions with sufficiently small or large dimensionless numbers (i.e. Lundquist numbers of $10^3 - 10^5$ are not so different than 10^{10} since the resistive terms in MHD will still be orders of magnitude smaller than the Lorentz terms). Previous work at Caltech [32] demonstrated that a concentric pair of planar electrodes can mimic the boundary conditions and evolution of a magnetically driven

plasma jet launched from an accretion disk system. Collimation due to strong $\vec{j} \times \vec{B}$ forces and the convection of toroidal magnetic flux [33] results in a singly line-tied screw pinch that can attain aspect ratios of $\sim 10:1$ before large current driven instabilities twist and break the column [73].

Kruskal and Shafranov were the first to identify the condition for current driven kink instability in a cylindrical plasma utilizing the definition of the safety factor [74]

$$q(a) = \frac{2\pi a B_z(a)}{L B_\theta(a)} < 1 \tag{4.1}$$

where a is the radius of the cylinder and L is its length. Increasing aspect ratio and/or jet current (both of which occur during the evolution of the Caltech jet experiment) results in a current driven kink instability which destroys the jet. Axial shear flows were predicted to stabilize such current driven instabilities in a Z-pinch configuration [36], and laboratory observations confirmed the theory by producing stable collimated jets with aspect ratios >30:1 with fixed length, fixed end-points, and no poloidal magnetic field [37]. Despite observing large ($\sim 50 - 70$ km/s) axial flow velocities on the Caltech jet, the setup presumably did not establish shear flows within the column sufficiently strong to stabilize against kink instabilities. As a result, the Caltech experiment was unable to replicate the large aspect ratios observed in many astrophysical jets.

4.2.1. Scaling relations

Since magnetohydrodynamic and two-fluid plasmas models have no intrinsic scale, establishing a sufficient scaling transformation should allow laboratory astrophysical jet experiments to reproduce jet dynamics that occur on astronomical scales. If such jets obey MHD scaling laws [71], then times should scale as $t_1/t_2 = a\sqrt{b/c}$, velocities as $v_1/v_2 = \sqrt{c/b}$, and magnetic fields as $B_1/B_2 = \sqrt{c}$, where $a = r_1/r_2$ is the ratio of jet radii, $b = \rho_1/\rho_2$ is the ratio of mass densities, and $c = p_1/p_2$ is the ratio of plasma pressures. Because astrophysical jets formed by active galactic nuclei generate relativistic flows, a direct laboratory model would require relativistic MHD plasmas. Smaller scale jets, such as those formed around young stellar objects (YSOs) therefore allow for a more reasonable parameter range that can be accommodated by laboratory experiments. These jets still exhibit significant parameter variation, and for many values only a

rough estimate is available. Table 4.1 presents jet parameters complied from a number of surveys [75, 76] compared to the parameters expected in the MoCHI.LabJet experiment for hydrogen jets.

Table 4.1
Comparison of Experimental and Astrophysical jet Paramters

Parameter	Labjet Design	YSO jet
Radius [cm]	$2-10$	$\sim 10^{15}$
Density [g cm^{-3}]	$10^{-7}-10^{-12}$	$10^{-18}-10^{-20}$
Temperature [eV]	$1-10$	$0.5-100$
Pressure [Pa]	0.8 - 8×10^{4}	$\sim 10^{-9}$
Magnetic Field [T]	$0.01-0.5$	$\sim 5\times10^{-7}$
Time Scale [s]	$10^{-6}-10^{-4}$	$10^{11}-10^{13}$
Velocity [km/s]	30-100	100-500

The ratio of jet timescales, velocities, and magnetic field strengths:

$$\frac{t_{YSO}}{t_{labjet}} \approx 10^{15}-10^{19}\,; \quad \frac{v_{YSO}}{v_{labjet}} \approx 1-17\,; \quad \frac{B_{YSO}}{B_{labjet}} \approx 10^{-6}-10^{-4} \tag{4.2}$$

overlap with the scaling parameters:

$$a\sqrt{\frac{b}{c}} \approx 10^{12}-10^{18}\,; \quad \sqrt{\frac{c}{b}} \approx 10^{-4}-10^{2}\,; \quad \sqrt{c} \approx 10^{-7}-10^{-4} \tag{4.3}$$

with sufficient fidelity to indicate that the MoCHI.LabJet experiment can replicate astrophysical jet dynamics on a laboratory scale.

4.3. Triple electrode planar plasma gun

Our experimental setup [77] uses the simplest feasible design that mimics the boundary conditions of an accretion disk around a magnetized central engine. The experiment (figure 4.1) uses three concentric, planar, and annular electrodes threaded by a vacuum dipole poloidal magnetic field $\vec{B}_{vac}(r,z)$. Independent biasing of the inner two electrodes with respect to the outer electrode provides control of the radial electric field profile $E_r(r)$. This design approximates an accretion disk rotating azimuthally in a poloidal magnetic field and provides a control knob on the azimuthal rotation profile $u_\phi \sim -E_r/B_z$. The two concentric cathodes also drive different poloidal currents $\vec{j}_{pol}(r)$ which provides some degree of control [33] on the axial flow shear $u_z(r)$. We also pay careful attention to providing an azimuthally symmetric mass source by injecting the working gas (hydrogen typically) through four annular slits in the electrodes. Symmetry at the footpoint of the jet is necessary for the plasma to rotate freely, unhindered by a discrete number of gas holes. This triple electrode design represents the next logical step up from previous planar plasma gun experiments that operated with two electrodes and discrete gas injection holes [73].

The custom plasma gun was built in house and consists of four sub-assemblies: the inner electrode assembly, the middle electrode assembly, the outer electrode assembly, and the high-voltage connection assembly. The inner electrode assembly consists of two electrically connected and concentric tube assemblies (1" and ½" outer diameter) that terminate into two annular copper electrodes (6cm diameter disc and 7.25cm diameter cup). Neutral gas is injected through four ¼" compression tube unions at the base of the assembly and becomes azimuthally symmetric before emerging from the slit between copper disc and cup at a radius of 3.05cm. The inner electrode assembly is attached to the middle electrode assembly through a high-voltage 2.75" ceramic break with an o-ring quick disconnect feedthrough.

The middle electrode assembly also consists of two concentric sub-assemblies that terminate into an annular copper electrode with an outer diameter of 17.8 cm and a 2mm gas injection slit at a radius of 4.4 cm. Gas is injected through twelve ¼" stainless steel tubes directly to the back of the copper electrode, making radial and azimuthal transits from the discrete injection points to fill up a plenum before exiting through the gas slit. The total assembly is mounted on a 10" CF flange on the vacuum chamber end-dome door through a high-voltage ceramic break. A bias magnetic field coil, consisting of 4mm square magnet wire wound 11×9 times around a 6" diameter phenolic support (1 mH, 1.5Ω) is mounted behind the electrode in between the two

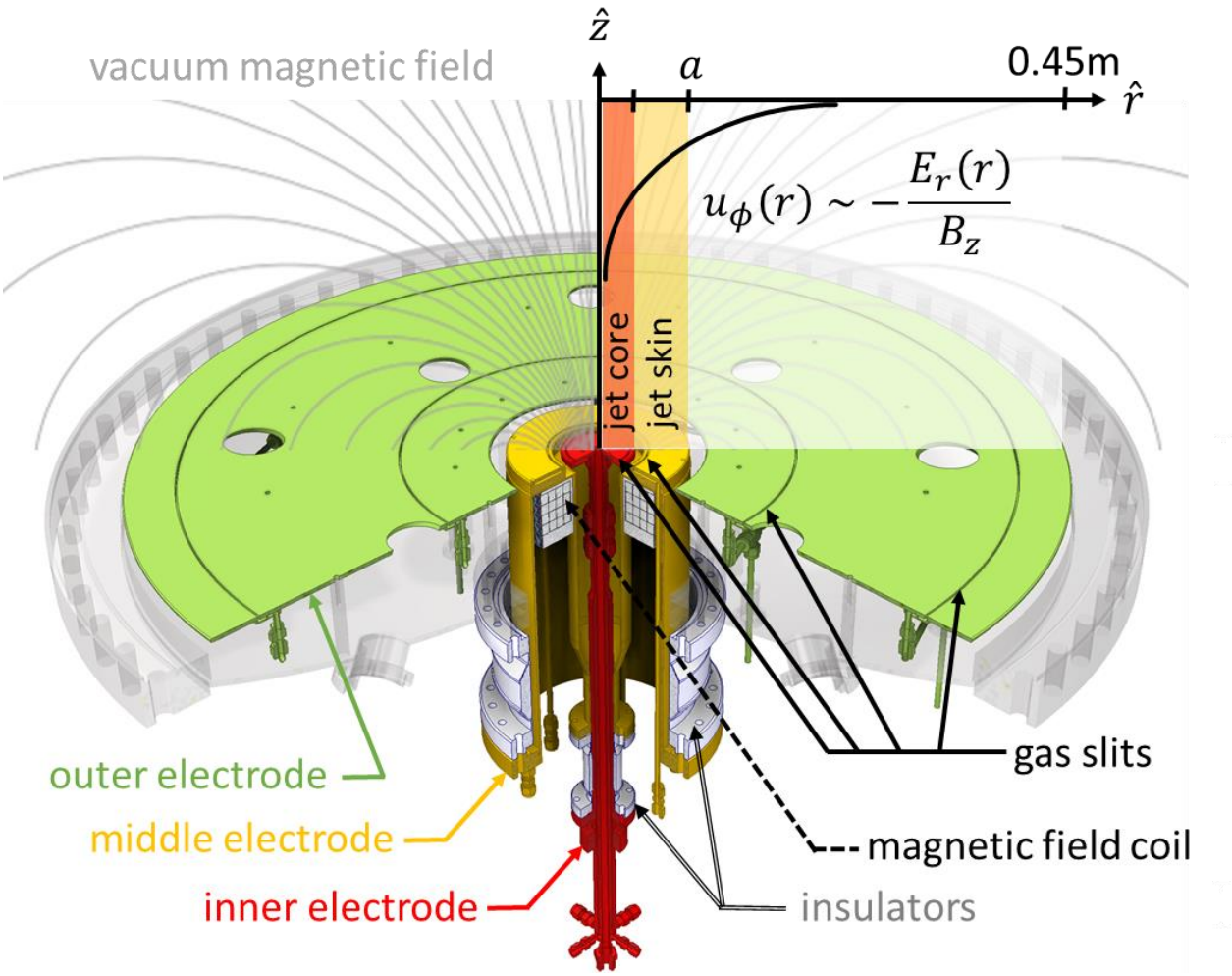


Figure 4.1. Concept behind the Labjet configuration of the mochi experiment: The setup replicates the basic configuration of an accretion disk rotating around a magnetized central engine with three concentric electrodes threaded by a poloidal magnetic field. The inner electrode (red) is designed to drive the core of the plasma jet and the middle electrode (yellow) to drive the outer skin of the jet, and therefore provide some degree of control on the jet current and flow profile. The mass sources are four azimuthally symmetric gas slits at the electrodes, allowing for unhindered azimuthal rotation.

concentric sub-assemblies.

The outer electrode is a copper anulus with a 90cm outer diameter and 19.5cm inner diameter. It consists of three 4.83mm thick copper sheets backed by two stainless steel gas injection plenums, providing an “outer-inner” and “outer-outer” gas injection slit at radii of 19.1 cm and 40.1 cm respectively. Each slit is fed by four ¼” stainless steel tubes which pass into the vacuum chamber through O-ring feed throughs before splitting (into 8 and 16 channels respectively) prior to reaching the plenums. The design of the plenum forces gas to zig-zag long

distances azimuthally and radially before entering the chamber as an azimuthally symmetric plume. Eight elliptic cutouts provide a circular projection along viewports on the end-dome door of the vacuum chamber. More detailed schematics and description of the gun design and fabrication can be found in reference 78.

4.4. Vacuum and fueling system

The complete LabJet gun assembly is mounted on a hinged end-dome of a custom, electro-polished, stainless steel vacuum chamber with a double o-ring seal. The vacuum chamber (Figure 4.2), manufactured to specifications by MDC Vacuum, is 1.4 m in diameter and 1.1m in length. Such a design, in theory, keeps the conducting walls far from the central plasma column and provides the boundary conditions relevant to astrophysical jets in nature. Ports include two 14" CF flanges at the north and south poles to house a Sumitomo Marathon CP-12 cryopump which provides 3600 L/s pumping rates to achieve a base pressure of $\sim 3 \times 10^{-7}$ torr without baking or surface conditioning, six 8" CF glass viewports along the equator aligned to the origin of the spherical vacuum chamber, thirty 2.75" CF glass viewports along the tropics and in front of each end-dome that point radially in a vertical plane (azimuthal plane) for diagnostics, sixty-four 3.375" CF glass viewports regularly distributed pointing towards the origin, and one 41" wire-seal port to mount a second planar plasma gun (MoCHI.SpiderLeg – the renamed DRX gun on loan from Los Alamos National Laboratory [79]). This second planar plasma gun is a double electrode gun with 8 pairs of discrete gas injection holes which serves as a comparison for the MoCHI.LabJet gun to highlight the influence of three electrodes and azimuthally symmetric gas injection slits over two electrodes with discrete gas holes. The vacuum chamber and cryopump is roughed by an Edwards XDS-10 scroll pump. If additional pumping is required (i.e. for the case of Helium which is not effectively pumped by the cryopump) a turbopump can be mounted to one of the 8" CF ports.

Gas fueling is provided by ten custom high throughput gas valves built in house based on valves inherited from Los Alamos National Laboratory to accompany the SpiderLeg gun (Figure 4.3). The valves feature sub-millisecond opening times and high gas throughput that is beyond the capability of commercially available valves. The valves operate by driving a pancake solenoid which repels an aluminum valve hat causing the valve to open. Spring compression and back line pressure provide a restoring force to the valve hat to close the valve after firing. The valves are

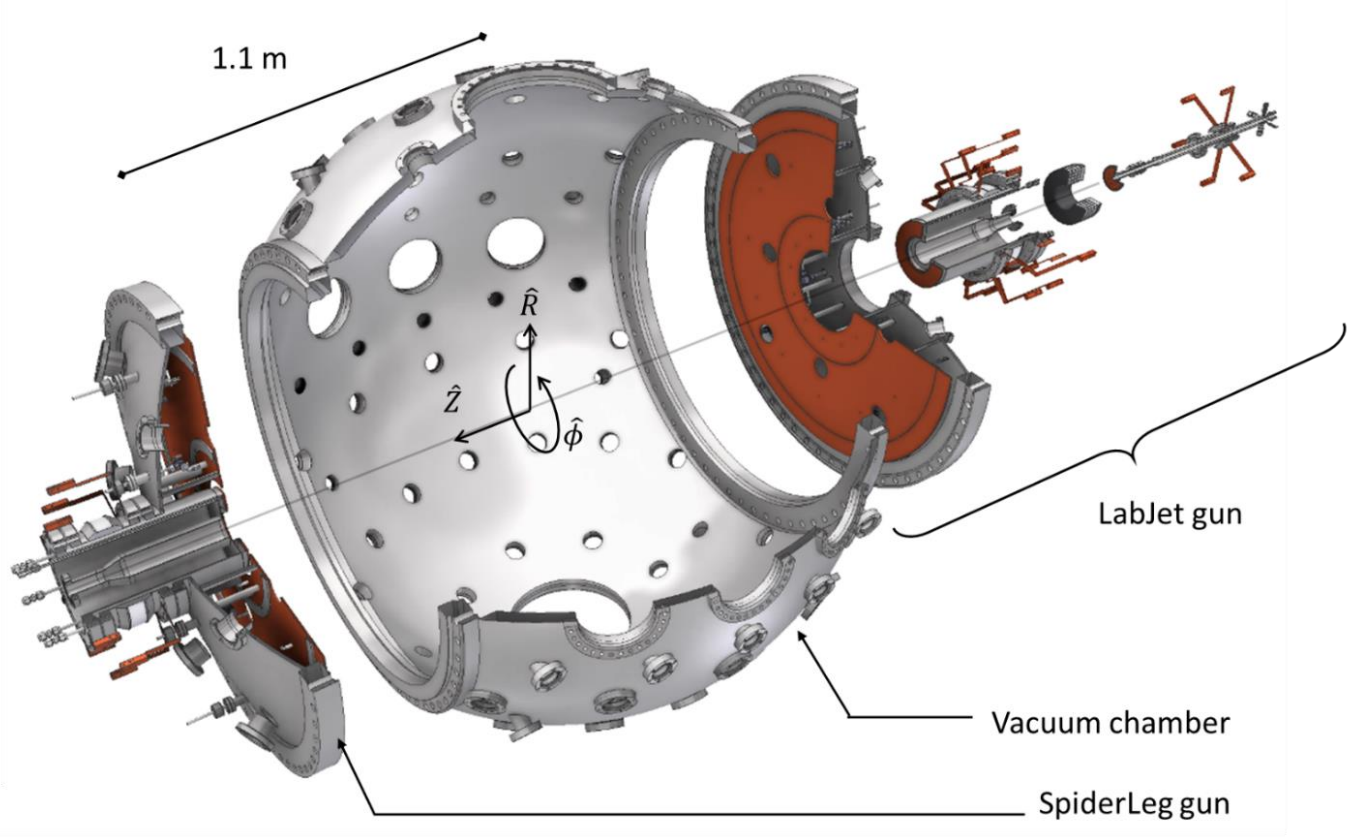


Figure 4.2. Full LabJet experimental setup with the spiderleg gun mounted opposite to the LabJet gun on the 1.4m diameter spherical vacuum chamber. Numerous viewports provide diagnostic and spectroscopic access and are arranged to point both radially inward as well as orthogonal to the $\hat{z}$ axis.

connected to a custom gas switchboard which allows operators to quickly and easily change the gas type used in the experiment. Four valves provide quadrant-controlled injection into the outer-outer and outer-inner gas injection plenums while the remaining two valves inject gas into the middle and inner electrode plenums. With all valves firing simultaneously at a voltage of 1.1kV, the total particle inventory injected into the chamber was determined to be $(7.8 \pm 0.6) \times 10^{23}$ particles for hydrogen with a backline pressure of 100 psi. This corresponds to a total neutral density of $(6.8 \pm 0.6) \times 10^{22} m^{-3}$ over the whole chamber [86].

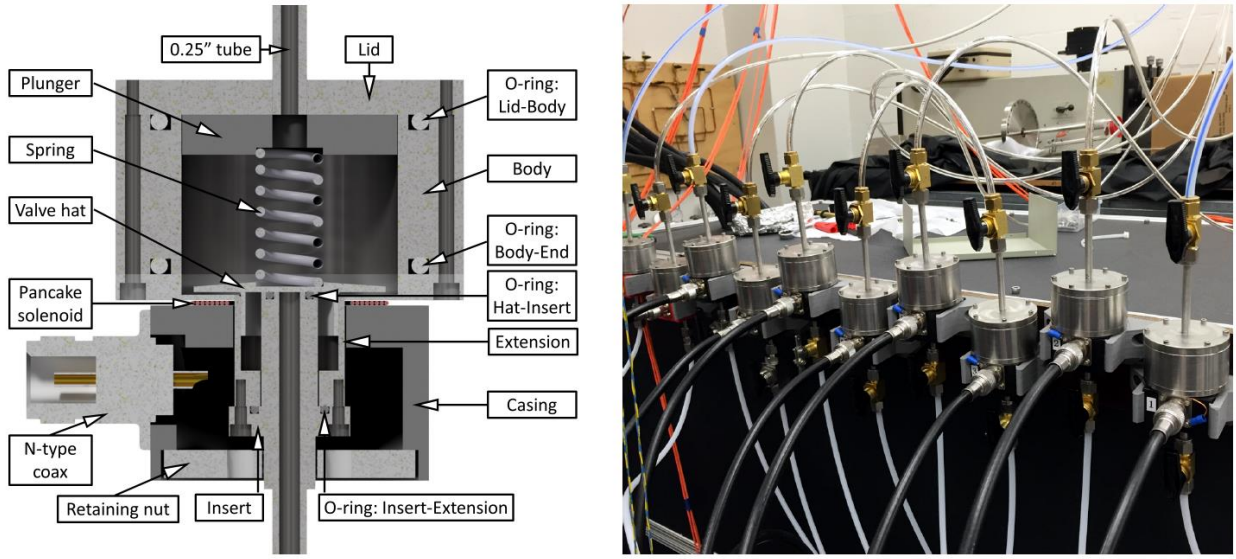


Figure 4.3. Left) Cutaway of custom fast gas valve. Right) Complete array of 10 custom fast gas valves, each powered and triggered independently via a 10-channel variable voltage power supply. Reproduced from [86].

4.5. Power Supplies

The LabJet gun is powered by two identical pulse power supply units that were designed and built in house [78]. Each PSU contains five capacitors (GA model 33838, $120\mu F$, 10 kV, 6 kJ each) connected in parallel with 1" wide silver-plated copper bus bars. The bus bars can be adjusted to vary inductances and produce a pulse-forming network [80]. The power supplies are labeled PSU2 and PSU3 and control the middle and inner electrodes respectively (Figure 4.3). High speed, high current switching is provided by custom co-axial ignitron towers that are cost effective and have low inductive impedance. The ignitrons are triggered through a transformer by a custom fast drive circuit [81]. Each PSU is charged with a Spellman SL10N300 HV power supply capable of driving 30mA of current at -10 kV. The power supplies are connected to the LabJet gun from the top of the ignitron tower with eight parallel coaxial HV cables and high-power RC snubbers (7.2Ωm, 0.05 μF). A similar, 12 kJ, $240\mu F$, $-10kV$ power supply (PSU1) can be used to drive the spiderleg gun (Ref [63] appendix c).

To power the bias magnetic field coils of the LabJet gun and the spiderleg gun, two power supplies are available. The first is a 12kJ, 600mF, 200V supply that consists of $600 \times 1mF$

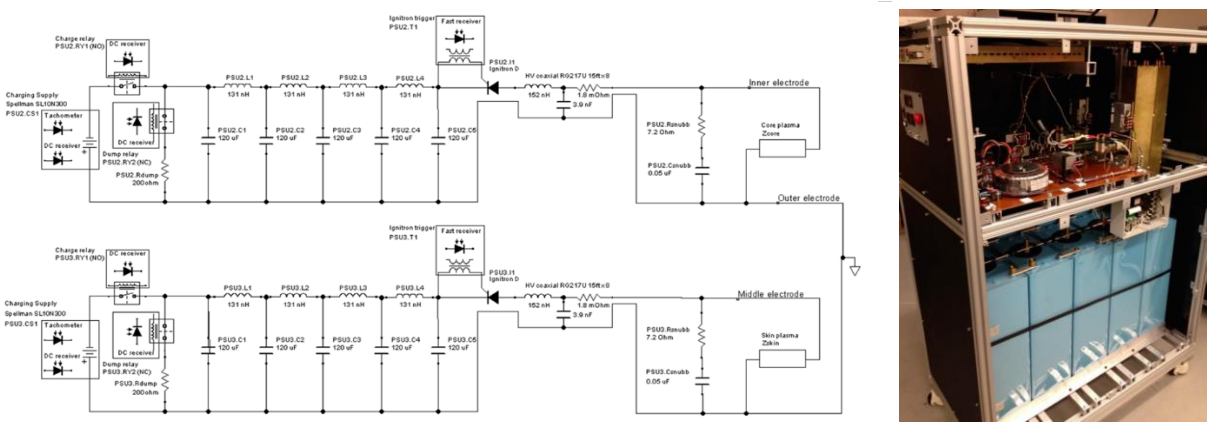


Figure 4.4. left) Circuit diagram of PSU2 and PSU3, which control the middle and inner electrodes respectively. More detail regarding the design, fabrication, and optically isolated control of the power supplies can be found in reference 78. Right) Picture of PSU 2 showing the 5 capacitors, internal circuitry and custom brass ignitron tower.

electrolytic capacitors connected in parallel. The charging supply is a TDKLambda Gensys 750W capable of driving 2.5A at 300V. The second power supply was inherited from Los Alamos National Laboratory and stores up to 4.5 kJ at up to 450V with a total capacitance of 48mF. Both power supplies are capable of driving several hundred Ampères (600 and 850A respectively) through the copper LabJet Bias magnetic field coil with a rise time of $\sim$20ms and $\sim$10ms respectively. The RLC pulse is critically-damped and is roughly constant on the $\sim 100 \mu s$ timescale of the plasma discharge. The strength of the field can be adjusted by adjusting bank voltage or the firing time of the coil with respect to the plasma discharge.

Gas injection into the LabJet gun is controlled by a 10-channel variable voltage power supply that allows for independent control of firing time and gas throughput for each channel (Figure 4.5). The supply was built in house and each channel consists of two $25 \mu F$ 1500V capacitors in parallel, charged through variable 50W resistors with a maximum resistance of 100kΩ by a LKB2197 power supply. Adjusting the position of the variable resistors allows for independent control of the capacitor voltage, and therefore control of gas throughput. Firing time for each channel is controlled by optotriggers which open a mosfet that sends a 24V pulse through a hand-wound ferrite core transformer to open an SCR switch.

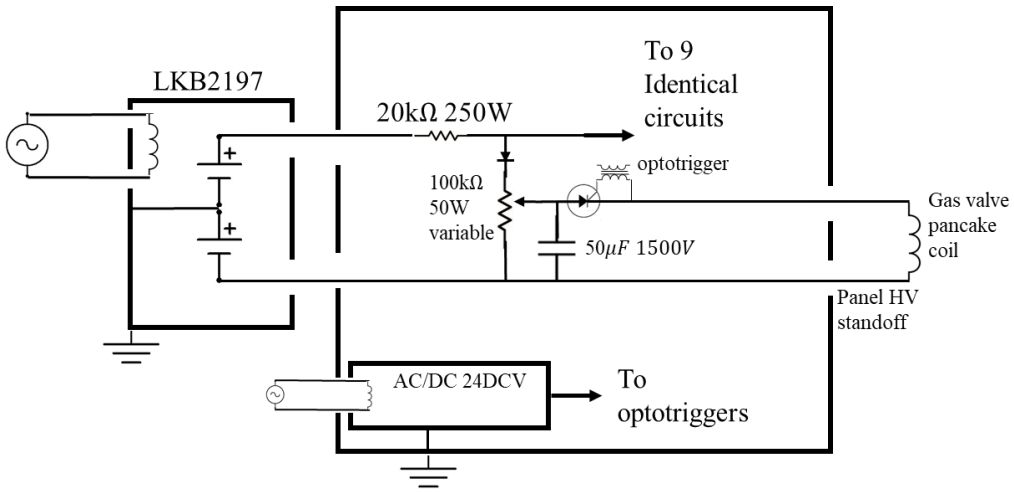


Figure 4.5. Simplified circuit diagram of the 10 channel, variable voltage power supply for control of the custom fast gas valves. Each channel is can be charged to a different voltage by adjusting the position of the variable resistor terminal. Firing times are controlled by 10 optotrigger circuits.

4.6. Control and data acquisition

Presented here is a summary of the MoCHI.LabJet control and data acquisition systems, a full description can be found in reference 63. For safe, convenient user operation, and to avoid ground loops, the experiment is controlled remotely with optical signals sent from a custom transmitter array connected to two National Instruments PXI-6602 timing cards with 16 pulse generators clocked at 80MHz and three PXI-6133 multifunction IO cards with 24 digital DC outputs. The optical signals are of three types: long-duration (DC) signals that control relay actuation, frequency-modulated pulses (0-400 Hz) that adjust bank charging voltages [78], and fast ($< \mu s$) single pulses that trigger the equipment and diagnostics in the desired sequence. Feedback signals (for measurement of bank voltages and currents) can also be optically transmitted back to digitizers by pulse-width modulation [78]. Fast optical pulses are converted into an electrical TTL pulse for triggering capacitor bank switches and diagnostics using a custom fast receiver with a switching time of 250 ns.

The data acquisition system has 120 digitizer channels, 96 of which consist of three NI-5752 analog-digital adapters mated to PXIe-7962R field-programable gate array cards programmed in-house for control of the data streams [82]. The FPGA's sample at 50 MHz with a 12-bit resolution. Although the theoretical data throughput is 3.2GB/s, onboard DRAM limited the throughput to 3GB/s with a total acquisition time of 800μs. The FPGA digitizers are dedicated

to the magnetic (Bdot) probe arrays to provide sufficient time resolution. The remaining digitizer channels are 24×14 bit, 2.5 MHz analog inputs on the PXI-6133 cards. An additional 8 measurement channels (not directly incorporated into the control infrastructure) consist of two Tektronix 2024B digital storage oscilloscopes that take 2500 samples at 2GHz. These oscilloscopes are dedicated to HV probes for measuring gun voltages and the outputs of a HeNe interferometer and are controlled wirelessly via remote desktop connection from a control computer.

A flexible GUI interface, MochiControl [83], allows operators to manipulate all experimental settings in real time, load previous shot settings, and store the acquired data in a hierarchical MDSplus tree [84]. The experiment control application accesses the storage through the MDSplus LabView interface [85], which leverages object-oriented and multi-threaded features of MDSplus. This allows the code to simultaneously translate LabVIEW and MDSplus data types and read and write to and from multiple MDSplus tree nodes. For each experimental shot MDSplus creates a new tree which can store numbers, strings, data arrays, and 2D data/time conglomerations called signals. This experimental data is stored together with the experimental shot settings to facilitate interpretation and shot reproducibility. Java based programs, Traverser and Jscope, provide access to the MDSplus trees and can present the stored data in a graphical format. A python module can also be installed which allows users to access data stored in the MDS plus trees.

The typical control sequence for a plasma shot proceeds as follows (LANL bias coil PSU used):

1. Close Bias coil charging relay and set to desired voltage using control knob, ensure correct firing time is entered in MochiControl
2. Ensure gas valve power supply is on and each channel is charged to desired voltage with correct gas valve firing timings entered in MochiControl
3. Ensure proper digitizer channels are active (including Tektronix scopes)
4. Toggle dump relay open on PSU2 and PSU3
5. Disable Spellman internal lock (HV arm) on PSU2 and PSU3
6. Enable high-voltage out on PSU2 and PSU3
7. Set charging voltage on PSU2 and PSU3 (Vdemand)
8. Toggle charge relay closed on PSU2 and PSU3 (charging begins)

9. Toggle charge relay open on PSU2 and PSU3 (charging complete)
10. Disable charging voltage (Vdemand) to protect tachometer circuit during discharge
11. Fire (initiates firing of all fast triggers set in MochiControl in the specified order at the specified timing)
12. Close dump relay on PSU2 and PSU3
13. Disable high-voltage out on PSU2 and PSU3
14. Enable Spellman Internal lock
15. Save shot data to MDSplus

4.7. Diagnostics

Several diagnostics have been acquired and constructed for the MoCHI.Labjet experiment. Standard diagnostics for a pulsed power plasma experiment include two Tektronix P6015 1000× high voltage probes connected to a four channel Tektronix oscilloscope for measurements of the middle and inner electrode voltages during an experimental shot. Hand wound, passively-integrated Rogowski coils calibrated to a commercial Pearson current monitor are used to measure the total and inner LabJet currents (and therefore the middle electrode current through the difference of the two signals). A Pearson current monitor is used to measure the current through the bias magnetic field coil.

Injected neutral gas density (prior to plasma breakdown) is measured using a custom fast-ionization gauge with $< 2\mu s$ rise time (appendix A). The gauge is used to optimize gas injection timings to prevent pre-fill and ensure azimuthally symmetric gas injection. During a plasma shot, the gauge is moved far away from the plasma and turned off to protect the electronics. Future modifications to protect the gauge during a plasma shot could allow tracking of neutral gas density in the chamber during a discharge. A standard Bayard-Alpert ionization gauge and thermocouple gauges provide measurements of the base chamber pressure. During a plasma shot, line integrated electron density measurements are obtained using an unequal pathlength He-Ne interferometer [86], where the two quadrature components of the phase angle difference are digitized using a Tektronix oscilloscope.

To measure the magnetic fields within the plasma, three 1-meter-long Bdot probe arrays (figure 4.6) have been constructed. These probes consist of a high-resolution center section

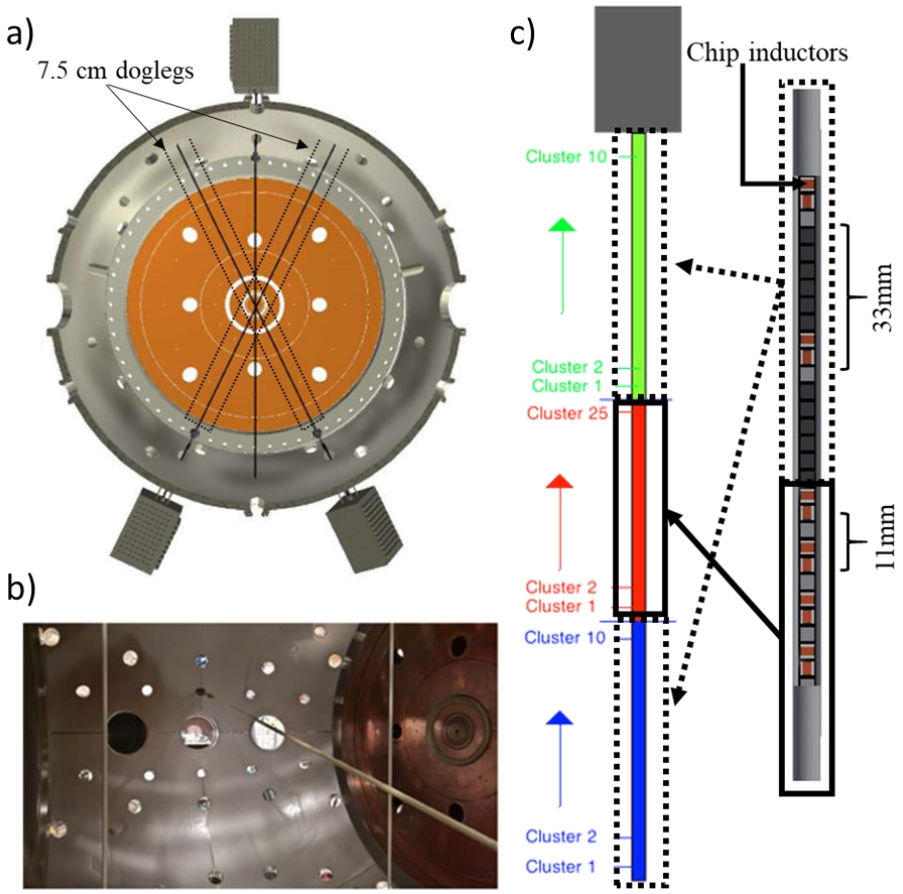


Figure 4.6. a) End on view of one possible magnetic probe array configuration, two arrays have a 7.5 cm dogleg and can be rotated into and out of the plasma column. b) Three arrays installed at different axial positions. c) Layout of Bdot probe array which consists of a high resolution central section with 25 clusters of 3 orthogonal inductor chips bordered by two low resolution sections with 10 clusters.

containing 25 clusters of three orthogonal inductor chips (Coilcraft 1008CS-472XGLB, $2mm \times 2mm \times 3mm$) spaced 11mm apart bordered by two low resolution sections consisting of 10 clusters spaced 33mm apart. The inductor chips are mounted in a Delrin stalk inside a 6mm OD stainless steel electrostatic tube shield inside of an 8mm OD alumina sheath. Two of the magnetic probe arrays have a 7.5cm dogleg bend allowing rotation into and out of the central plasma column. The signals are carried to the FPGA digitizers by double-shielded, twisted-pair CAT-7 cables. A custom, Pi-type, load-balanced, switchable 96 channel attenuator printed circuit board attenuates the signal amplitudes by 0-76 decibels in increments of 6, 10, and three 20dB. The circuit is load balanced so that capacitive and inductive pickup rejection is further enhanced

by the common mode rejection of differential digitization. The magnetic probes are calibrated to account for orientation errors and frequency response.

Optical line spectra are measured using a 1m focal length f/8 monochromator (Horiba Yvon 1000M) with a 2400 line/mm diffraction grating coupled to a dual inline frame Princeton Instruments PI-MAX3 1024×1024 ICCD camera (Figure 4.7a,b). The camera has a fast phosphor decay time and inline pixels which allows for two frames to be taken in a plasma discharge, typically with an exposure time of $\sim 1 \mu s$ and interframe time of 3-15μs. The camera can also be separated from the monochromator to take high-speed photographs of the plasma jet. Multichannel ion doppler spectroscopy is made possible by a 96-channel, 10m long optical fiber bundle (figure 4.7d, appendix C) constructed in house from 105μm core multi-mode fibers, arranged in three vertical lines of 32 fibers at the matching-optics side of the spectrometer (Figure 4.7b, c). The bundle breaks out into twelve separate bundles of 8 fibers arranged in a linear array at the plasma collimating optics side. The complete bundle is extremely cost effective at about $27 per channel, mainly due to savings realized in the bundle breakout and optical mounts, designed and printed in house with an off-the-shelf 3D printer, collimating lenses sourced from laser pointer parts, and protection tubing sourced from local hardware stores. The twelve bundles can be mounted on the vacuum chamber CF ports with custom 3D printed mounts.

A final diagnostic available to the MoCHI.LabJet experiment is a Kirana fast-frame CMOS camera manufactured by Specialized imaging. The camera can take 180 sequential 924×768 10-bit frames at up to 5MHz and is made available through the University of Washington Student Technology fund. During operation, the Kirana was typically set to 1-2MHz with exposure times of 500ns and 200ns respectively. The corresponding time window is 180 and 90 μs respectively, providing a full video of the jet evolution which takes place over $\sim 80 \mu s$.

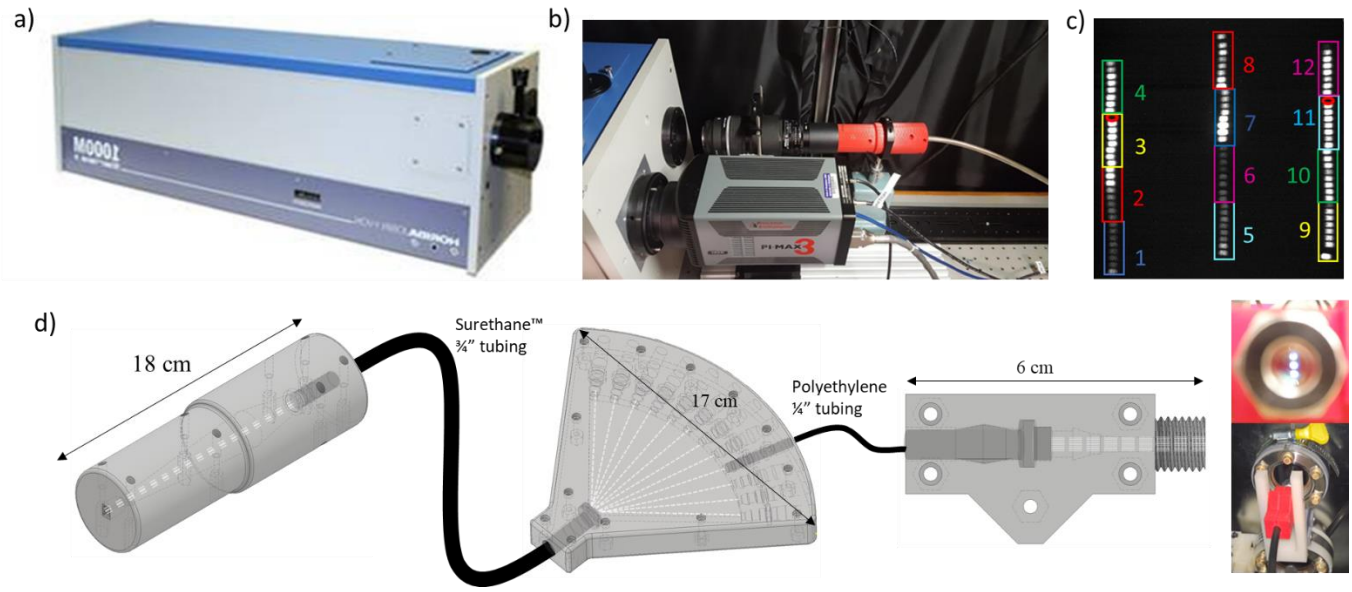


Figure 4.7. a) Horiba 1000M f/8 spectrometer. b) Matching and collecting optics (back to back Nikon 35, 105mm lenses and a PI-MAX3 ICCD camera. c) CCD view of optical fiber bundle backlit with hydrogen reference lamp, individual bundles have been labeled. d) Custom 10m long optical fiber bundle, consisting of a 3×32 channel v-groove chip clamp, 12-way fiber breakout, and 12×8 channel v-groove clamps with laser diode collimating lenses.

4.8. First plasmas

First plasmas were achieved in November 2015 while the commissioning phase of all hardware was nominally completed by August 2016. The goal of the commissioning phase was to test all major power supplies, control hardware, data acquisition systems, and to optimize jet performance. Primary diagnostics during this phase were limited to current and voltage probes and the high-speed PI-MAX 3 ICCD camera. Initial operation was characterized by breakdown behind the plasma gun and between the high-voltage insulating gap separating the middle and outer electrodes. As a result, current intended to drive the jet was driven elsewhere and jets were either unable to form or were short lived and uncollimated. Fine tuning of the gas timings resulted in the first observation of a flared plasma jet when operating with the middle electrode only (middle and outer-inner gas injection slits) at -2.5 kV (Figure 4.8) with peak currents of $\sim 20 kA$. At this point, the outer-inner and middle gas injection slits were each fed by a custom gas valve (Figure 4.3). Only three custom valves were initially available, and so to provide additional gas injection for all four gas injection slits, four Parker series-9 valves powered by custom fast valve drive units were used to replace the single custom valve on the outer-inner gas injection slit. Testing with the Parker

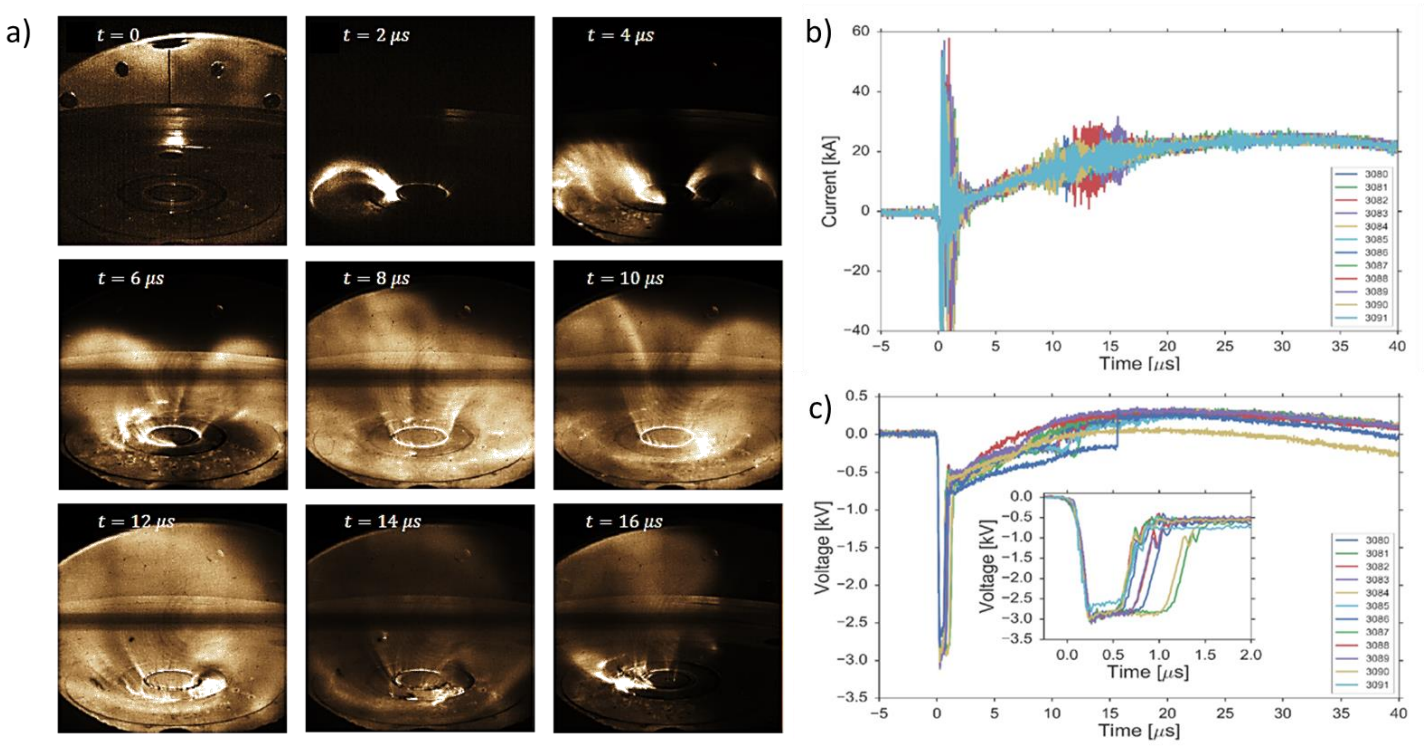


Figure 4.8. a)Time evolution of flared plasma jet constructed from shots 3080-3091 by stepping camera acquisition times by $2\mu s$. Shots were conducted using diatomic hydrogen with the middle electrode only at -2.5kV. Arcing across insulating gap appears to disrupt plasma jet development b) Current traces for shots 3080 – 3091 reveal total jets currents of $\sim$20kA. c) Voltage traces for shots 3080-3091, the initial drop at $t = 0$ reveals when the gun ignitron is fired, breakdown occurs shortly afterward at $t \approx 1\mu s$. Figures adapted from reference 78.

valves revealed that they had an insufficient throughput and opening times to effectively promote breakdown between the middle and outer-inner gas injection slits (Fig 4.9). These results motivated the development of 10 additional custom fast-gas valves and an accompanying 10 channel variable voltage power supply (figures 4.3 and 4.5).

Completion of the custom fast gas valve array marked the start of the first operational campaign. Collimated jets were observed when operating the inner electrode at a voltage of -4 kV. Growth of a current driven kink instability was observed to depend on the strength of the bias magnetic field for a given jet current (Figure 4.10). This observation was in qualitative agreement with the Kruskal-Shafranov instability condition (equation 4.1), which can be expressed as [87]

$$\lambda_{gun} > 4\pi/L \tag{4.4}$$

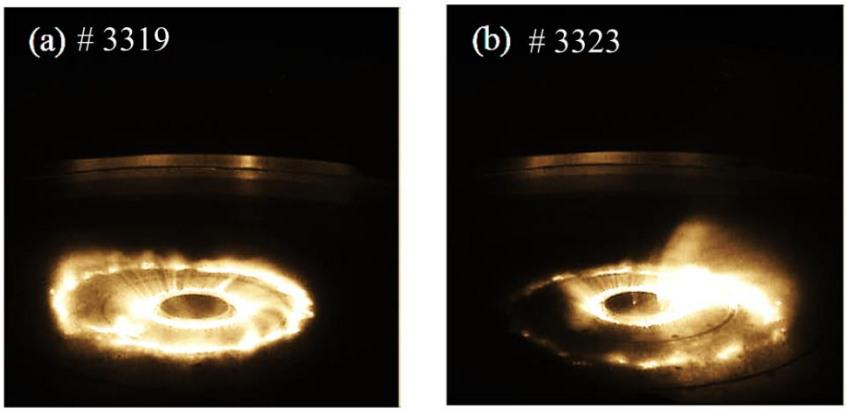


Figure 4.9. a) Azimuthally symmetric breakdown during the 'bundt cake' formation phase when operating the middle electrode only. Gas injection is provided by a custom valve to the middle gas injection slit and four parker valves to the outer-inner gas injection slit. Images reveal that the outer footpoints do not terminate at the outer-inner gas injection slit as expected. b) Similar breakdown occurs when no gas is injected into the outer-inner slit. These observations suggest insufficient gas is being provided to the outer-inner slit.

where $\lambda_{gun} = \mu_0 I_{gun}/\psi_{gun}$ can be thought of as a boundary condition imposed on the plasma gun at the surface (I_{gun} is the total current and ψ_{gun} is the bias magnetic flux). Combined operation of both electrodes (or middle electrode alone) continued to suffer from erroneous arcing behind the gun and between the high voltage insulating gap between the middle and outer electrodes. To prevent this arcing, a 1/16" thick sheet of PTFE Teflon plastic was rolled around the middle electrode re-entrant port and allowed to protrude $\sim 1cm$ in front of the electrodes. This modification was completed alongside the development and calibration of an unequal pathlength heterodyne interferometer for electron density measurements [86].

With these additions, the first respectable plasma jets were obtained while firing both inner and middle electrodes simultaneously. Figure 4.11 outlines the typical evolution of the plasma jets observed during this phase of operation. After fully charging the capacitor banks, the bias coil is triggered with the timing and voltage adjusted to provide the desired level of bias magnetic flux (Figure 4.11a). Then gas is injected via ten custom fast-gas valves with quadrant controlled injection into both slits on the outer electrode and single valves controlling injection into the middle and inner electrodes (Figure 4.11b). Typical gas valve timings were -3 to -4 ms with respect to ignitron firing times for hydrogen. When sufficient neutral gas density was present in

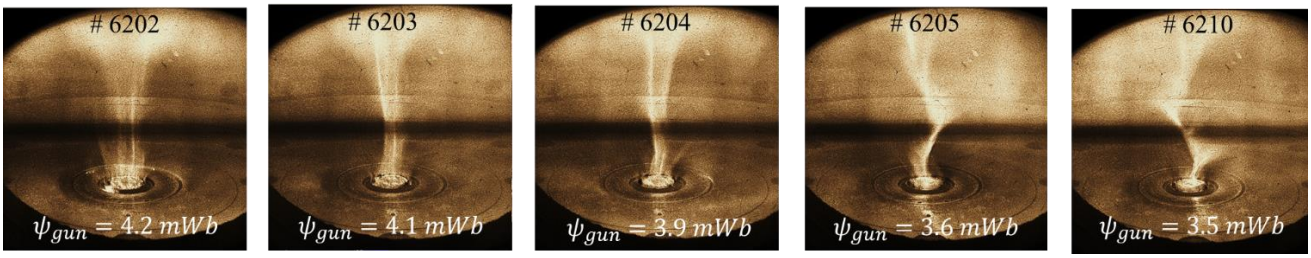


Figure 4.10. ICCD images of various plasma shots (inner electrode only) taken at the same time and jet current for different vacuum magnetic field strengths. The development of a current driven kink instability is observed to depend on the strength of the bias magnetic flux ψ_{gun}. The kink occurs more rapidly at lower bias magnetic fields (larger $\lambda = \mu_0 I_{gun}/\psi_{gun}$), in qualitative agreement with the Kruskal-Shafranov instability condition (equation 4.4).

front of the gun electrodes, the ignitrons of each bank are simultaneously fired ($t = 0$) resulting in breakdown along magnetic field lines linking the inner and outer-outer gas slits (core current) and middle and outer-inner gas slits (skin current) (Figure 4.11c). This initial formation is an azimuthally symmetric version of the discrete 'spider leg' arches observed in the Caltech experiment [32] and was dubbed the 'Bundt cake' formation due to the physical resemblance between the nested arches and a Bundt cake. As the current increases, the nested arches expand and zipper up along the central axis (Figure 4.11d). Strong axial flows driven by an MHD pumping mechanism [33] convect toroidal magnetic flux from the base of the jet where the radius is small to the end-tips where the radius is large, resulting in an increased pinch force which acts to collimate the jet (Figure 4.11e). Azimuthal flows at the jet footpoints are also observed due to $\vec{E} \times \vec{B}$ rotation which is unhindered by the continuous gas injection slits (Figure 4.11c-g). As the current ramps up and the jet lengthens and collimates, the plasma column quickly becomes susceptible to current driven kink instabilities which cause the jet to break at a critical aspect ratio and propagate downstream as a detached plasmoid (Figure 4.11f-g). Significant amounts of energy remaining in the capacitor bank then go into driving a diffuse secondary plume, where bright arches are also observed to form over the Teflon insulator (Figure 4.11g).

A characteristic discharge (#6580) with middle and inner electrode voltages of -4 and -4.6 kV respectively and a bias magnetic flux of $\sim 3.4 mWb$ was filmed using the high speed Kirana camera at 2Mfps (figure 4.12a). The total jet current during the initial, collimated jet ramps up to $\sim$60kA (with $\sim$20kA through the inner electrode) and achieves peak jet current of $\sim$175 kA

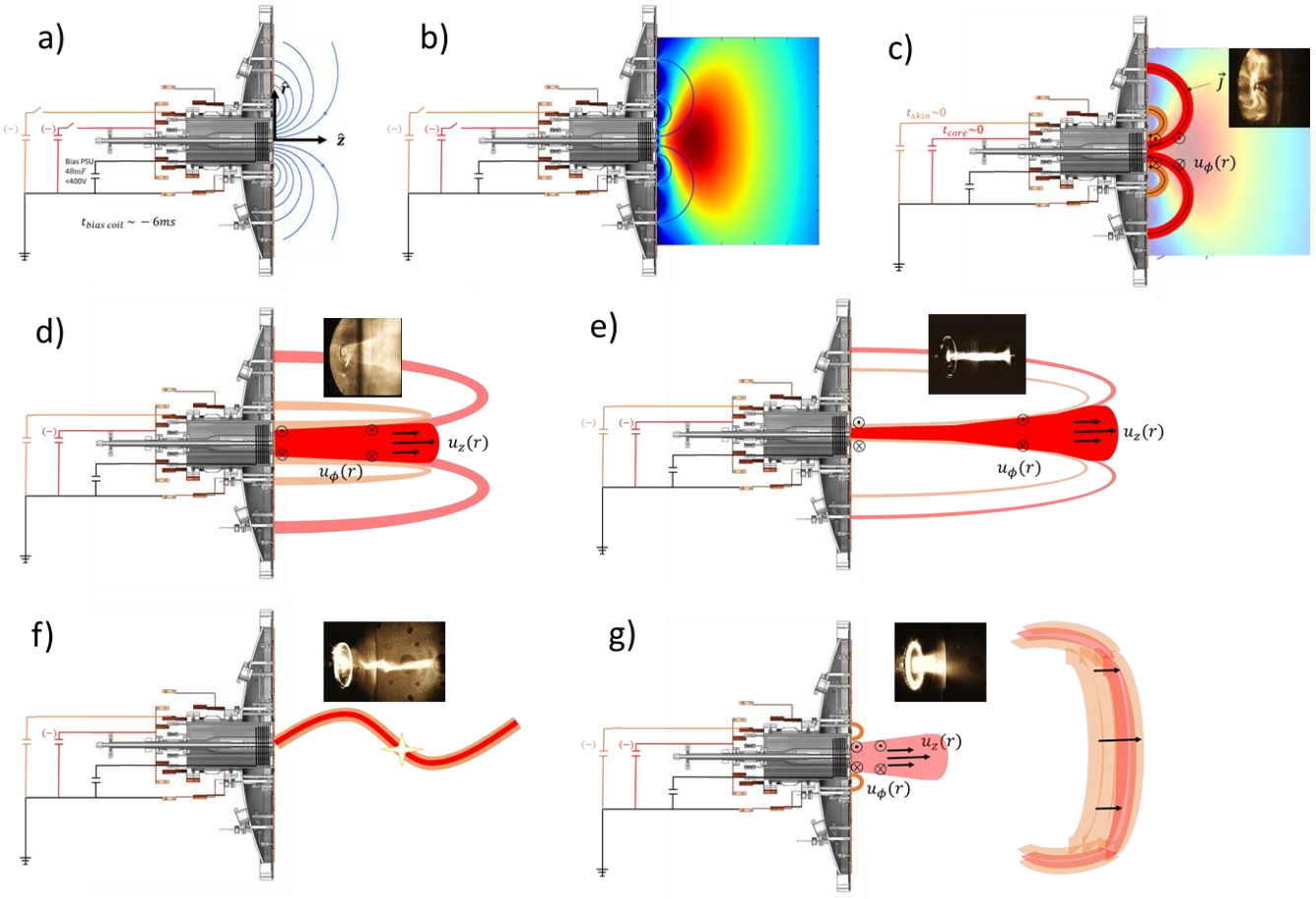


Figure 4.11. Typical operation sequence for first operational campaign with Teflon insulator between middle and outer electrodes. a) Vacuum bias magnetic field is applied. b) Neutral gas is injected by 10 custom fast gas valves to form an azimuthally symmetric plume in front of the electrodes. c) Ignitrons are fired and breakdown occurs along magnetic field lines linking the inner and outer-outer gas slits (red core-current) and middle and outer-inner gas slits (orange skin-current). This formation phase is termed the 'Bundt cake' due to the physical resemblance between the nested arches and a Bundt cake. An azimuthal flow due to $\vec{E} \times \vec{B}$ forces develops. d) The jet zippers up along the central axis, driving axial shear flows and convecting plasma from the base of the jet. e) Convection of toroidal magnetic flux leads to collimation of the plasma column. f) Jet reaches critical aspect ratio for a given jet current and bias flux, resulting in a kink instability which breaks the jet. g) Initial jet propagates as detached plasmoid and a secondary diffuse jet forms, bordered by bright arches over the Teflon insulator.

at $t \approx 40\mu s$ (with ~85kA through the inner electrode) during the diffuse secondary jet (Figure 4.12b). Line integrated electron density measurements [86] (figure 4.12c) show that the plasma density rises rapidly during the initial jet to $\sim 8 \times 10^{20}\ m^{-2}$. Based on the inferred jet diameter of ~4cm this corresponds to a local electron density of $\sim 2.5 \times 10^{22} m^{-3}$. Following the onset of the kink instability (Figure 4.12a frame 4) the electron density drops precipitously before gradually increasing during the formation of the secondary diffuse plume.

Focusing on the evolution of the primary jet (figure 4.13), high speed video provides a first estimate of the axial velocity (figure 4.13b) by tracking the visible light emission with a known calibration factor between number of pixels and distance (figure 4.13a frame 7). Such an approach assumes that the light emission is representative of the density of the plasma column, which appears to be a reasonable assumption based on the correlation between line integrated electron density measurements and camera images (figure 4.12a,c). This analysis reveals axial velocities of ~70 km/s which is significantly faster than the predicted sound speed but in good agreement with the Alfvén velocity for a hydrogen plasma with a density of $2.5 \times 10^{22}\ m^{-3}$ and a magnetic field strength of 0.5 T (predicted for 60kA of current through a ~5cm diameter plasma column). These results are in good agreement with the MHD pumping theory [33, 34] and corroborate observations from the Caltech jet experiment. Further agreement to previous laboratory jet experiments can be found through an investigation of jet evolution through classical stability space (figure 4.13c). As the jet lengthens and the current ramps up, the jet approaches the classical Kruskal-Shafranov instability condition (equation 4.4) and goes unstable. The kinked plasma column then undergoes magnetic reconnection as the kink instability couples to a Rayleigh-Taylor [88] or sausage instability [89] before propagating across the chamber as a detached plasmoid.

An estimate jet temperature during the evolution of the primary jet can be obtained through gun current and voltage traces, which indicate a total plasma impedance of $\sim 25\ m\Omega$ for the core plasma and $\sim 100 m\Omega$ for the skin plasma. Assuming Spitzer resistivity, this back-of-the-envelope estimate indicates that the core plasma temperature rises from ~5 eV to ~9 eV, while the skin starts colder at ~1 eV and increases to match the core plasma temperature at the time of detachment (figure 4.14). These calculations account for estimated plasma inductance $l(t) = B^2 \pi a^2 L/(\mu_0 I^2) \sim 30 - 80\ nH$ and inductive impedance $Z_L = 2\pi l/\tau \sim 4\ m\Omega$ for a magnetic field strength of $\sim 0.5T$, current $I \sim 60$ kA, and RLC ring period $\tau \sim 140\mu s$ and suggests that the total impedance of the cold plasma jet is dominated by plasma resistivity and not by plasma inductance. It is important

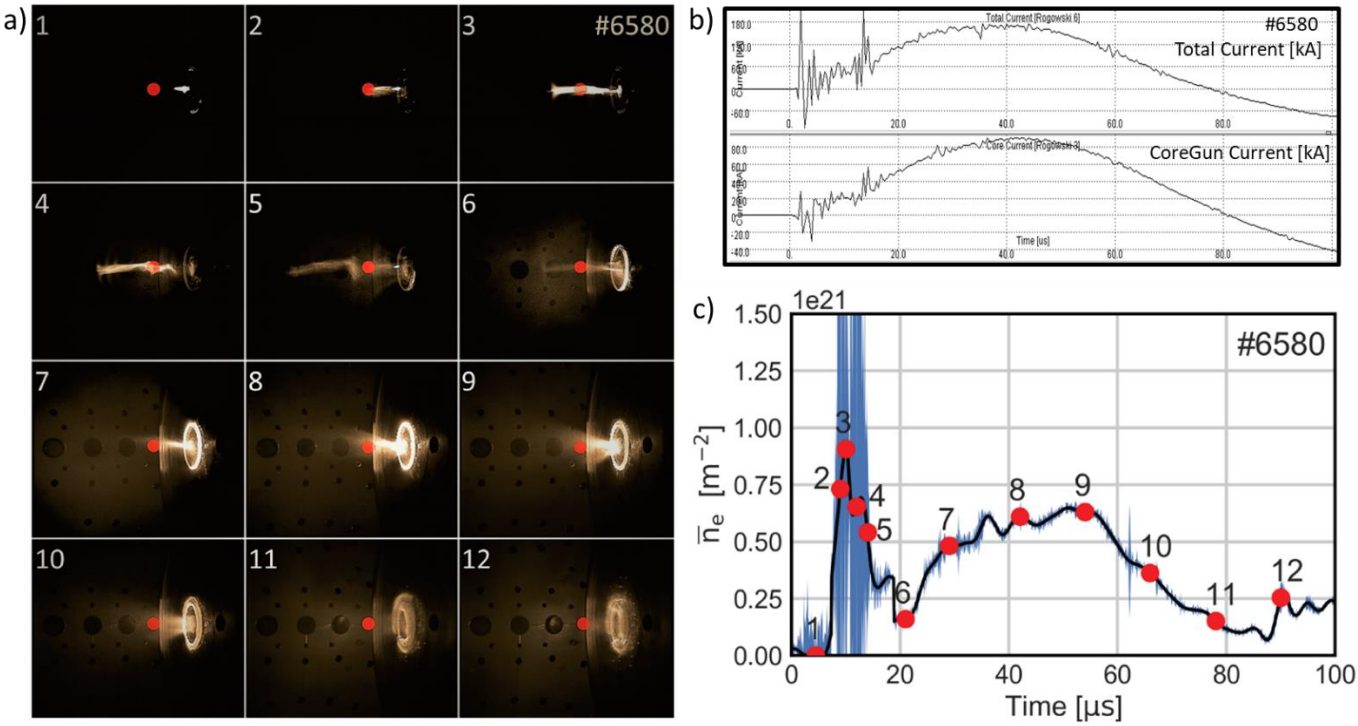


Figure 4.12. Characteristic discharge (#6580) during the first operational campaign of the LabJet experiment. ($V_{core} = -4.6kV, V_{skin} = -4kV, \psi_{gun} = 3.4mWb$) a) Select frames from high speed video taken with the Kirana camera. The red dot indicates the path of the He-Ne interferometer providing electron density measurements and each frame number corresponds to the labeled red dots in panel c. b) Gun currents measured using calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. Panels (a) and (c) reproduced from reference [86].

to note however, that this analysis neglects electrode sheath effects and other impedances.

The results of the first operational campaign informed an upgrade intended improve the performance of the secondary diffuse plume. At peak current, it was suspected that the bright arch over the Teflon insulator was draining current from the core of the jet and causing it to become more diffuse and less collimated. To prevent this the Teflon wrap was pushed further back into the middle electrode re-entrant port and an alumina end cap with a higher work function was Torr sealed to the 17.8 cm middle electrode diameter in the insulating gap. The insert protrudes ~1cm in front of the electrodes and extends far enough back that the Teflon (in theory) does not experience regions of high electric field. This addition marked the start of the second operational phase of the experiment between Dec. 2016 and Mar. 2017, and was characterized by significant

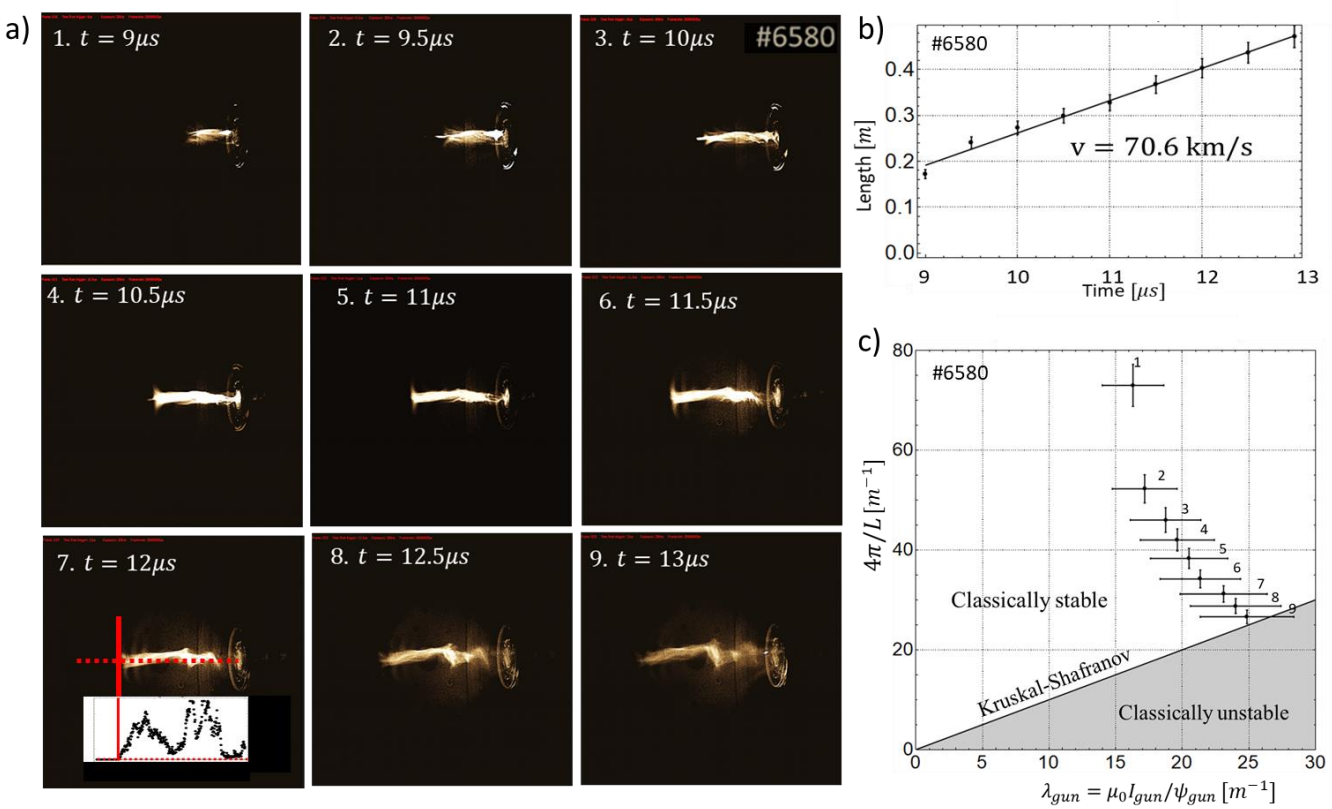


Figure 4.13. Evolution of the primary jet. a) High speed camera images taken with Kirana at 2Mfps, light emission is used to calculate the length of the plasma jet (frame 7). b) Axial velocity of 70.6 km/s is calculated by plotting the jet length as a function of time. c) Evolution of the jet through stability space indicates that jet goes unstable at the predicted Kruskal-Shafranov instability limit.

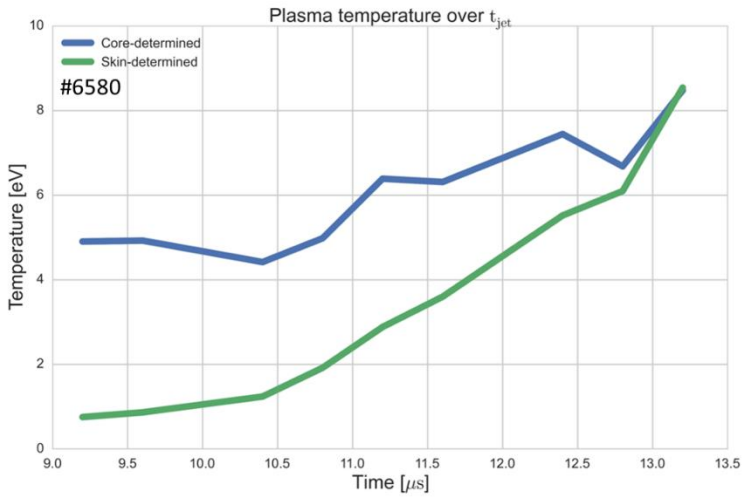


Figure 4.14. Estimate of temperature evolution within the primary jet assuming Spitzer resistivity. The blue line shows the core-determined temperature using the inner electrode voltage and current waveforms, while the green line shows the skin determined temperature.

improvement in the secondary jet performance. Although arcing over the alumina (reduced following electrode polishing) continued and resulted in significant shot-to-shot variation, three distinct phases were typically observed (Figure 4.15). Stage I continued to be the formation of a narrow, collimated jet (Figure 4.15 a-e Figure 4.16a frames 1-2, and Figure 4.17a frames 1-2) which lengthens as the current ramps up before going kink unstable near the predicted Kruskal-Shafranov instability limit (Figure 4.15f, Figure 4.16a frame 3, Figure 4.17a frame 3). Following detachment, a secondary diffuse plume forms into the plasmoid (Figure 4.15 g-h, Figure 4.16a frames 4-6, and Figure 4.17a frames 4-6). This stage II plume is transition period and is characterized by weaker electron densities (Figure 4.16c) which rise slowly as the current increases and the plume collimates and lengthens to reach the opposite end of the vacuum chamber 1.1m away. The trajectory in stability space (Figure 4.16d and Figure 4.17d) shows that the stage II plume forms in the classically stable region but crosses into the classically unstable region as the current ramps up to its peak value (Figure 4.16b and Figure 4.17b). This secondary plume appears to push the detached plasmoid, forming a double bow shock that is observed in camera images (Figure 4.17a frame 4) and line integrated electron density measurements when the laser path is in the central midplane (Figure 4.17c).

The third stage begins approximately when the gun current driving the core of the jet (inner electrode) becomes larger than the gun current driving the skin (middle electrode) (Figure 4.16b and figure 4.17b). This occurs just after peak total current and $\sim 40 - 50 \mu s$ after breakdown. At this point the secondary plume becomes a stable, collimated jet with an aspect ratio $\gtrsim 20{:}\,1$ (1.1m in length with $\lesssim$ 5cm radius based on optical emission) and electron densities comparable to the initial short, unstable jet ($\sim 10^{22} m^{-3}$) (Figure 4.16c and Figure 4.17c). This long jet remains stable for $\sim 40 - 60 \mu s$ corresponding to > 40 radial Alfvén transit times (where $v_a \approx 70 km/s$ has been assumed and later confirmed with magnetic probe measurements), despite occupying classically unstable space (Figure 4.16d, and Figure 4.17d). Temperature estimates assuming spitzer resistivity (figure 4.18) suggest that temperatures increase from $\sim 5 eV$ to $\sim 10 - 20$eV despite a decrease in the total current during this period. High speed video of these long, stable jets also suggests strong axial and azimuthal flows which may account for the unexpected stability. Other interesting features, such as pronounced helical structure can also be observed upon a more careful inspection of individual frames (Figure 4.19). To quantitatively investigate these features, a third operational phase between Apr.-Jun. 2017 began following the completion of a custom

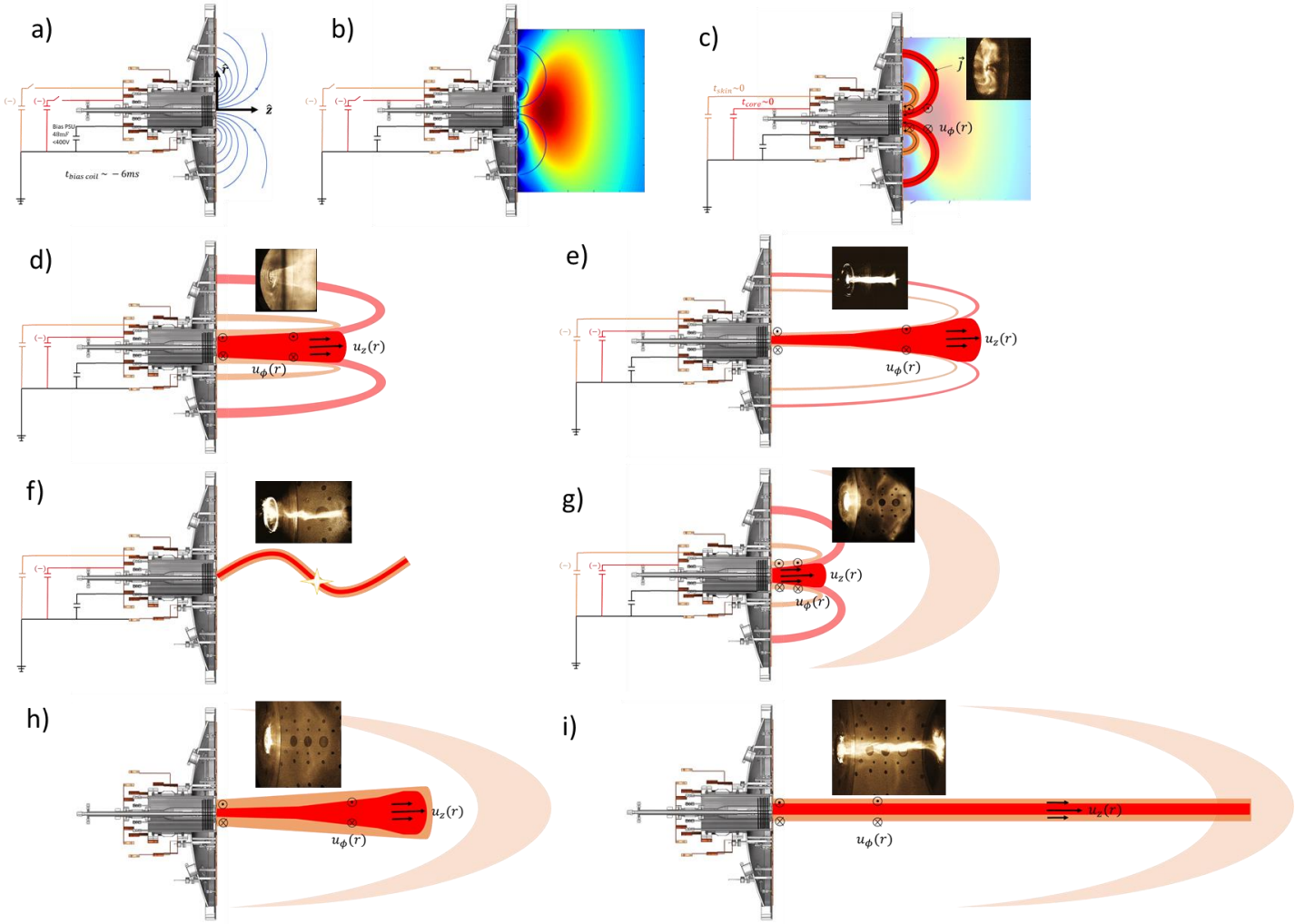


Figure 4.15. Typical operation sequence for second and third operational campaigns with alumina insert between middle and outer electrodes. a) Vacuum bias magnetic field is applied. b) Neutral gas is injected by 10 custom fast gas valves to form an azimuthally symmetric plume in front of the electrodes. c) Ignitrons are fired and breakdown occurs along magnetic field lines linking the inner and outer-outer gas slits (red-core current) and middle and outer-inner gas slits (orange-skin current). This formation phase is termed the 'Bundt cake' due to the physical resemblance between the nested arches and a Bundt cake. An azimuthal flow due to $\vec{E} \times \vec{B}$ forces develops. d) The jet zippers up along the central axis, driving axial shear flows and convecting plasma from the base of the jet. e) Convection of toroidal magnetic flux leads to collimation of the plasma column. f) Jet reaches critical aspect ratio for a given jet current and bias flux, resulting in a kink instability which breaks the jet. g) Initial jet propagates as detached plasmoid and a secondary diffuse jet forms. h) Secondary diffuse jet lengthens into the plasmoid. i) Secondary jet collimates into long, stable jet.

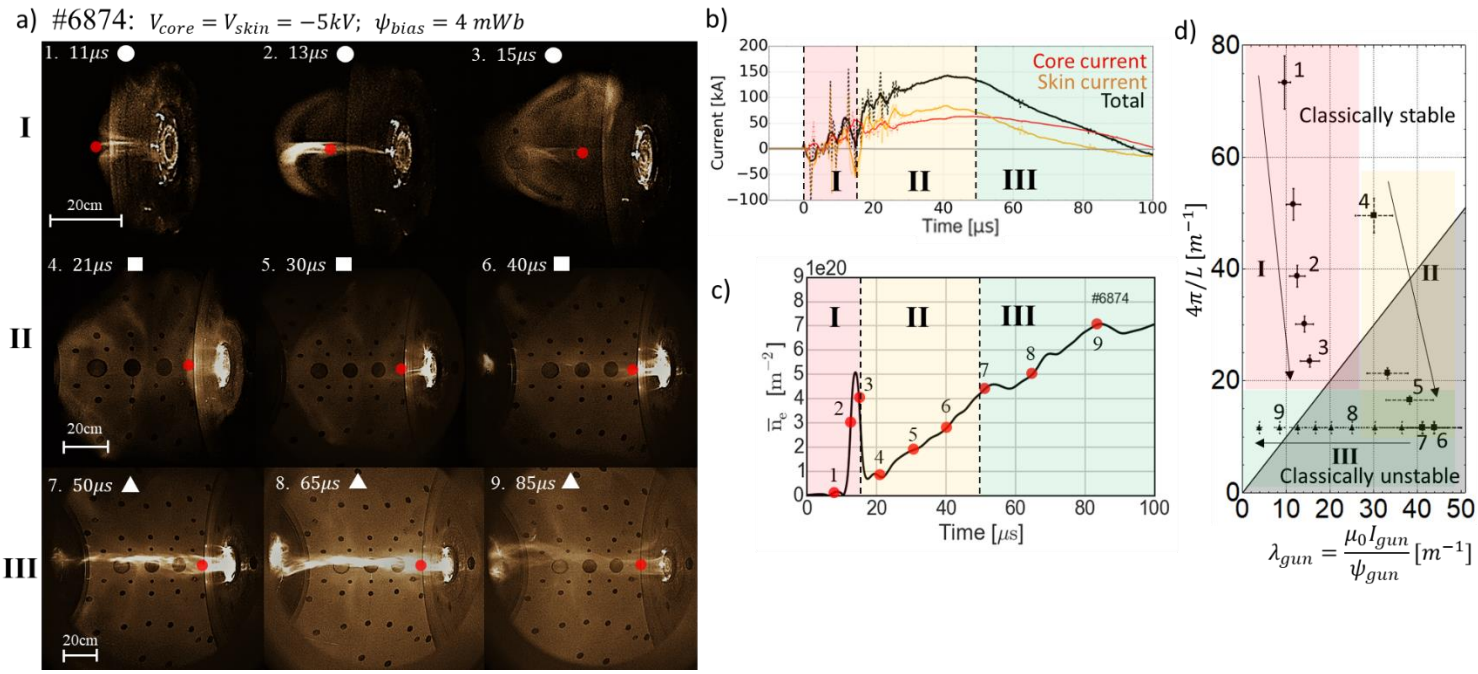


Figure 4.16. Characteristic discharge for the second operational campaign. a) High speed camera images taken with a Kirana camera at 1Mfps. The red dot indicates the path of the He-Ne interferometer and each frame number corresponds to the labeled red dots in panel c. b) Gun currents measured using calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density shows a spike for the initial jet and gradual rise during the formation of the long stable jet. d) Jet evolution through classical stability space. The initial jet lengthens and goes kink unstable near the Kruskal-Shafranov instability limit. The secondary diffuse plume forms in stable space and crosses into unstable space as the current increases. The long jet remains stable with a fixed length (1.1m) for tens of μs despite occupying classically unstable space.

multichord optical fiber bundle and three custom magnetic probe arrays. The following two chapters will discuss the implementation and results from these diagnostics.

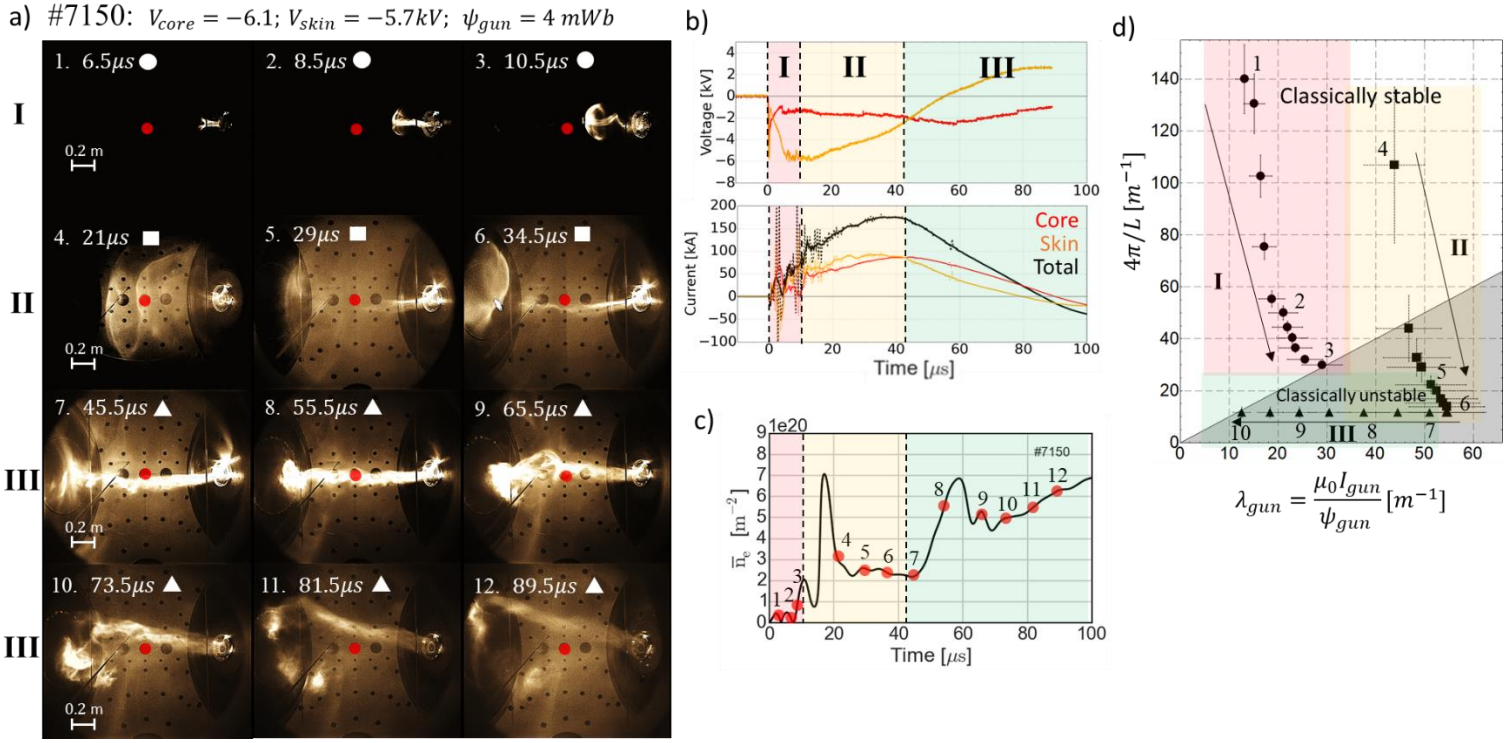


Figure 4.17. Characteristic discharge for the third operational campaign. a) High speed camera images taken with a Kirana camera at 1Mfps. The red dot indicates the path of the He-Ne interferometer and each frame number corresponds to the labeled red dots in panel c. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density shows a spike as the detached plasmoid propagates across the chamber and rise when the bowed long stable jet intercepts the laser beam. d) Jet evolution through classical stability space. The initial jet lengthens and goes kink unstable at the Kruskal-Shafranov instability limit. The secondary diffuse plume forms in stable space and crosses into unstable space as the current increases. The long jet remains stable with a fixed length (1.1m) for tens of μs despite occupying classically unstable space.

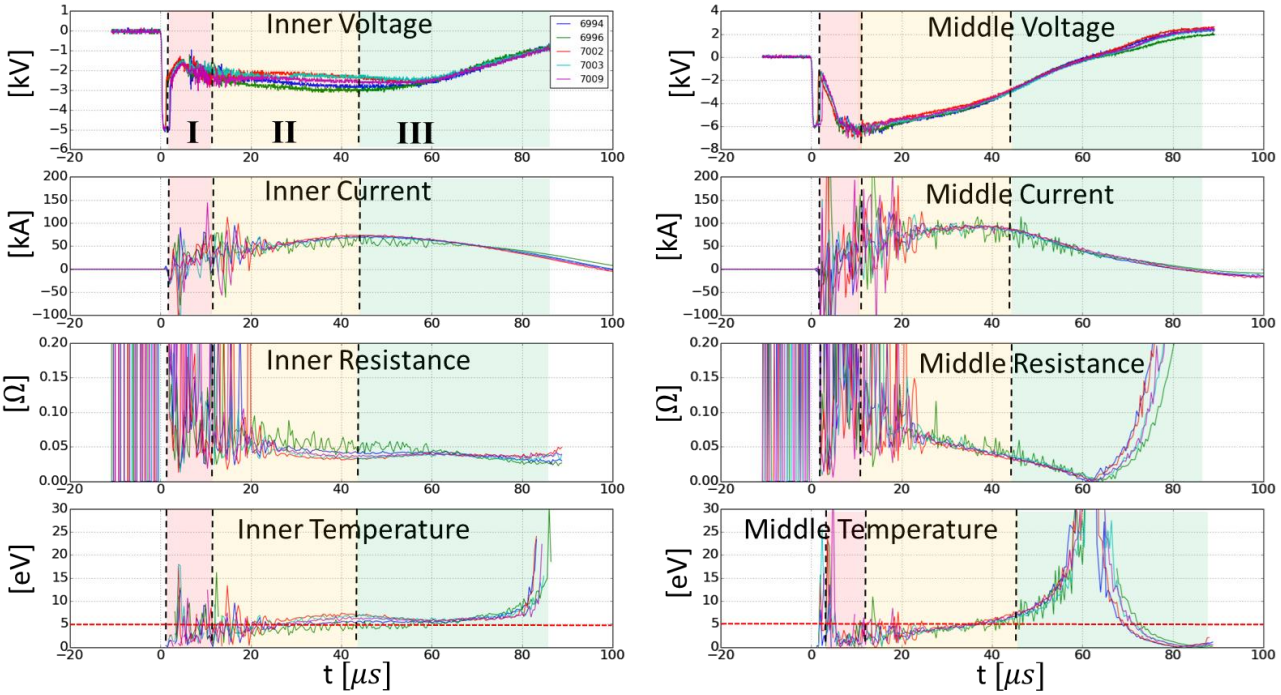


Figure 4.18. Temperature estimate from gun current and voltage waveforms assuming Spitzer resistivity. The waveforms are noisy during stage I and the start of stage II but are smooth during stage III, with a total resistance of $\sim 50 m\Omega$. Temperature estimates account for inductive impedance; however, this contribution is an order of magnitude less than the resistive impedance. At the start of stage III, the core and skin temperatures are ~ 5 eV with the skin increasing rapidly to > 20 eV at $t \sim 60 \mu s$ and the skin increasing more gradually towards the end of the discharge.

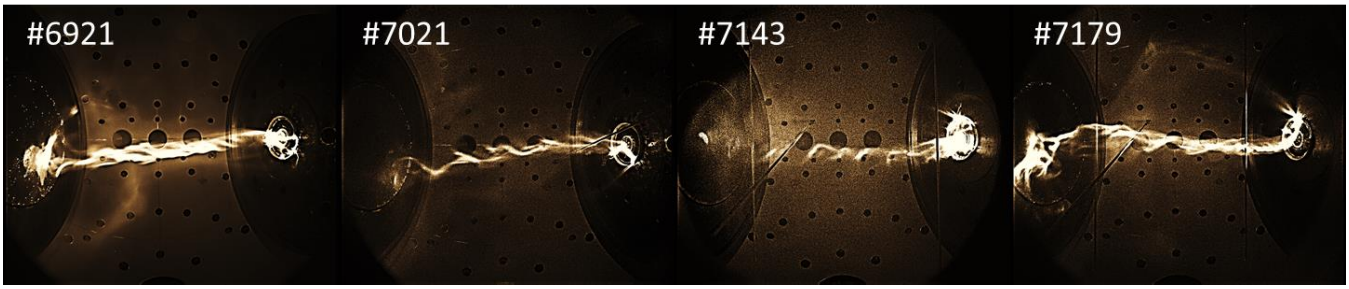


Figure 4.19. High speed camera images of several shots that exhibit a distinct helical structure. A similar structure is observed on most shots; however, it is often obscured due to saturation of the Kirana camera or, for the case of highly collimated jets, the helical winding is so tight that it is difficult to discern.

Chapter 5
ION DOPPLER SPECTROSCOPY MEASUREMENTS ON THE MOCHI.LABJET EXPERIMENT

5.1 Theory

Unlike most states of matter, plasma emits a variety of electromagnetic radiation ranging from Bremsstrahlung "braking radiation" due to electron-ion collisions, recombination radiation due to the capture of free electrons, and line emission due to excitation and relaxation of bound electrons. Due to the discrete difference in energy levels (i.e. quantization) of bound electrons, as they transition between states they emit or absorb a photon with specific energy $h\nu_{ij} = E_i - E_j$ where h is Planck's constant, ν_{ij} is the frequency of the emitted or absorbed photon, and $E_{i,j}$ are the energy levels of the transition. The discrete nature of this emission/absorption of line radiation makes it particularly useful and powerful as a plasma diagnostic tool.

In general, there are two types of plasma spectroscopy methods, active and passive. In active methods, plasma atoms are stimulated in some way (i.e. laser with a specified wavelength or neutral beam) while the emission or absorption of the specific line radiation is observed to infer various plasma parameters such as density, temperature, and velocity. Such approaches typically require significant experimental effort and are more difficult to implement. Passive spectroscopy on the other hand uses light emitted from the unperturbed plasma itself, where bound electrons are excited primarily due to electron impacts. The rate of spontaneous decay of the excited state is considerably faster than the timescales of interest in the plasma [90] and can be thought of as nearly instantaneous. While the emitted photons could, in theory be absorbed by another atom, this process is negligible in laboratory plasmas which are optically thin. The observed line radiation is never perfectly monoenergetic and broadening of the spectral line profile (in wavelength space) can be caused by various mechanisms including Doppler broadening, pressure broadening, broadening by electric fields (Stark broadening), magnetic fields (Zeeman effect), and of course broadening introduced by the spectrometer apparatus itself. Stark and Zeeman broadening are particularly important in very dense plasmas and in strong magnetic fields, respectively. Doppler broadening, on the other hand, is a particularly useful tool for inferring the temperature of a plasma (see below), and since it is stronger for light elements in the visible wavelength, Doppler

broadening of the Balmer H_α line at 656.3 nm is often used to infer hydrogen plasma temperatures with sufficient accuracy.

Although the wavelength of an emitted photon is invariant in the particles frame of reference (NIST database provides comprehensive list for various ions [91]), the observed central wavelength can differ substantially in the viewing frame due to Doppler shifts resulting from particle motion. The shift in central wavelength can be related to the particle velocity v along the line of sight by

$$\lambda_{measured} = \lambda_0 \left(1 + \frac{v}{c}\right) \tag{5.1}$$

where $\lambda_{measured}$ is the observed central wavelength, λ_0 is the emitted wavelength, θ is the angle between the velocity vector and the line of sight, and c is the speed of light. For a plasma in thermal equilibrium, the emitting particles will have a velocity distribution described by a Maxwellian (equation 1.2, simplified and reproduced here for convenience)

$$f(v) = \sqrt{\frac{m}{2\pi k_B T}} \exp\left[-\frac{m(v - v_{average})^2}{2k_B T}\right] \tag{5.2}$$

where m is the particle mass, T is the temperature, $v_{average}$ is the average velocity and k_B is Boltzmann's constant. Assuming that broadening of the spectral line profile is primarily due to Doppler temperature broadening (which produces a Gaussian line shape), the temperature of the plasma can be deduced from the from standard deviation σ of the gaussian spectral intensity profile

$$T = \frac{\sigma^2 c^2 m}{\lambda_0 k_B}. \tag{5.3}$$

Because the spectrometer itself contributes to the broadening of the spectral lines, it is important to account for the instrument function contribution when attempting to calculate the plasma temperature in this way. It is also important to note that because spectral intensities are measured along a chord through the plasma, the measured spectra (and thus calculated plasma parameters) are chord integrated values and can be affected by the optical depth of the plasma. For the plasmas and wavelengths used in this experiment, the plasma is considered to be optically thin and chord integrated measurements therefore traverse the entire plasma. Techniques to infer local plasma parameters from chord integrated measurements will be addressed in following sections.

5.2 Hardware

The ion Doppler spectrometer (IDS) is a 1 m focal length Horiba 1000m Czerny-Turner type monochromator with an $f/8$ aperture. Two interchangeable gratings, 2400 l/mm and 1800l/mm, are available but the 2400 l/mm grating was used preferentially due to the improved spectral resolution. The spectrometer is coupled to a PI-MAX-3 1024×1024 ICCD camera at the exit slit, which can take two frames per discharge with an interframe time limited only by the phosphor intensifier decay time (to 10% in $2\mu s$). Light from the plasma is gathered by a custom 10m long optical fiber bundle (appendix B) built in house from $105\mu m$ core multi-mode fibers. The fiber bundle consists of 12 bundles of 8 individual fibers which can be mounted in various arrangements on the vacuum chamber CF ports. Plasma side collimating optics consist of 4.6mm focal length bi-aspheric acrylic collimating lenses (one for each bundle) mounted inside of a Swagelok compression nut. The focus can be tuned by adjusting the distance between the fibers and the lens using a threaded 8 channel V-groove chip clamp. At the spectrometer side, the fibers are arranged into three linear arrays of 32 using three V-groove chips inside of a custom 3D printed clamp. To match the f-stop of the fibers to the f-stop of the spectrometer, two back to back Nikon camera lenses (35mm and 105mm) are used. To accommodate the physical size of the fiber bundle and maximize the field of view, the spectrometer entrance slit has been removed. This allows for fiber bundle dimensions of $\sim 5mm$ in the vertical direction (spatial direction) and $\sim 6mm$ in the horizontal direction (wavelength space) at a central wavelength of 656.3 nm. Reducing or Increasing the central wavelength of the spectrometer, reduces or increases (respectively) the horizontal image extent and sets a minimum central wavelength of $\sim 450nm$ to simultaneously observe all three fiber columns. The maximum central wavelength for the 2400 l/mm grating is 740nm.

5.2.1 Calibration

The spectral resolution of a Czerny-Turner monochromator can be calculated from the monochromator and ICCD parameters according to

$$R(\lambda) = \frac{\Delta x}{2L}\left(\sqrt{\left(\frac{2d\ cos\alpha_0}{m}\right) - \lambda^2} + \lambda \tan \alpha_0\right) \tag{5.4}$$

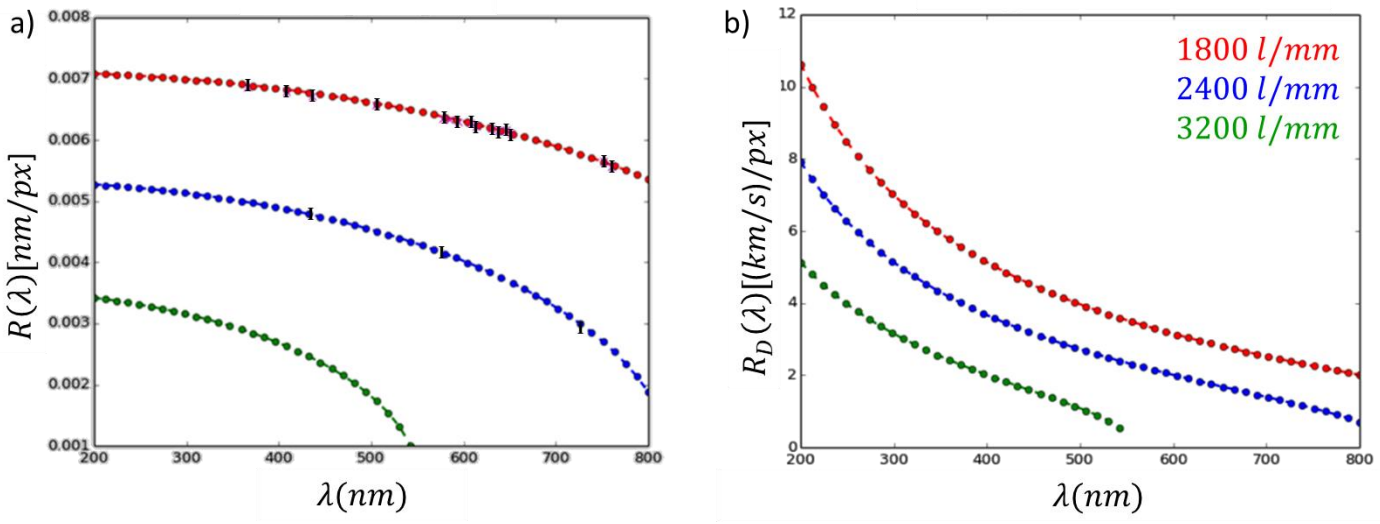


Figure 5.1. Theoretical dispersion curve (a) and Doppler resolution (b) for various gratings. The value $\alpha_0 = 5.25$ in equation 5.4 was found to provide the best fit to measurements taken with a spectral lamp of known doublets. For H_α emission at 656.3nm, a 2400 l/mm grating provides a Doppler resolution of ~1.6 (km/s)/px. For CII emission at 723.64nm, the same grating provides a resolution of ~1.25 (km/s)/px.

where $R(\lambda)$ is the resolution per ICCD pixel (nm/px), Δx is the pixel size of the ICCD, L is the focal length of the monochromator, α_0 is the half angle between the incident and diffracted light on the grating, d is the groove spacing, m is the diffraction order (typically set to unity), and λ is the selected wavelength of the monochromator. A derivation of this equation can be found in the appendix of reference 92. For the PI-MAX3 ICCD the pixel size is 12.8μm and using spectral lamps with known doublets (nearby line spectra with a known wavelength separation) $\alpha_0 = 5.25$ was found to provide the best fit to experimental measurements (Figure 5.1). The corresponding doppler resolution in (km/s)/px is

$$D(\lambda) = R(\lambda)\frac{c}{\lambda}. \tag{5.5}$$

With a known spectral resolution, calibration of the optical fiber bundle can be accomplished using spectral lamps by setting the spectrometer wavelength such that the calibration line is imaged to the same location on the CCD as the plasma line of interest. Holding a spectral lamp to the vacuum chamber window to backlight the fibers and taking a long exposure with the

PI-MAX3 ICCD therefore provides a reference image (Figure 5.2a) of the optical fiber bundle which accounts for relative offsets between fibers. This allows us to delineate a reference zero for each fiber to which we can compare our shot data. Calibrating the spectrometer in such a way is difficult for several reasons. First the grating adjustment is not perfectly accurate so any adjustment of the central wavelength requires a new calibration image to be taken. Furthermore, any bumps or vibrations of the fiber bundle, spectrometer, or collimating optics can misalign the fibers requiring a new calibration image to be taken. Finally, in some situations no satisfactory calibration lines could be found using the available spectral lamps. In situations where no satisfactory calibration lines could be found, long exposures ($> ms$) of a plasma discharge were taken with the assumption that the time integrated velocity along each line of sight averages to zero (Figure 5.2b,c). Such an approach, in theory, allows for calculation of the line integrated plasma velocity (shift in the centroid position) but does not easily permit calculation of temperature since the reference image is itself Doppler broadened.

To improve the signal to noise ratio, and because information in the spatial (vertical) dimension is not essential, each fiber in the CCD image is binned vertically within a region indicated by the red rectangles in figure 5.2. The resulting intensity profile for each fiber is then fit with a gaussian distribution

$$f(x) = \frac{A}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x - x_0)^2}{2\sigma^2}\right] + mx + b \tag{5.6}$$

where x is the horizontal location (in pixel space), A is the amplitude of the distribution, x_0 is the centroid, σ is the standard deviation, m is the slope of any offset, and b is the offset. In some cases, it was sufficient to set $m = 0$ as there was no significant slope in the background signal (Figure 5.2a for example). Binned intensity profiles from reference images taken using spectral lamps as well as long exposure plasma discharges are both well fit by the gaussian distribution. The centroid of the fits (x_{i0}) sets the refence zero velocity while the standard deviation σ_i for spectral lamp reference images is assumed to correspond to the instrument broadening and is used to calculate the instrument temperature ($T_{instrum}$) based on the mass and wavelength of the observed plasma emission. Gaussian fits in pixel space can be easily converted to wavelength space using the known spectrometer dispersion relation (equation 5.4) where x_{i0} of each reference fiber image corresponds to the known emission wavelength (λ_0) of the spectral line of interest.

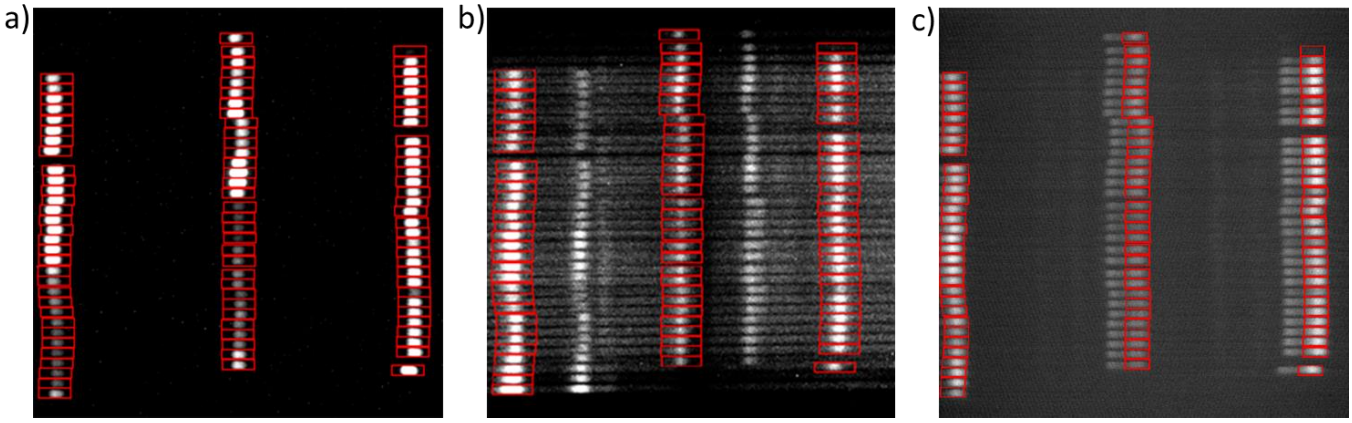


Figure 5.2. Reference images taken at various wavelengths. Rectangles indicate the binning volume in the vertical (spatial) direction. a) Reference image taken using a Hydrogen spectral lamp at 656.3nm. b) Reference image of CII impurities taken at a spectrometer central wavelength of 723.392 nm over a long exposure plasma shot. The two spectral lines are CII doublets at 723.132 nm and 723.642 nm (selected in red rectangles). c) Reference image of NII emission at 648.2 nm over a long exposure nitrogen plasma shot. Nearby NI emission can be seen to the left of the red rectangles (red shift in wavelength space).

5.3 Line integrated plasma parameters

For each shot, gaussian fits to the binned intensity are compared with fits to the reference image in order calculate the chord integrated plasmas parameters (Figure 5.3). The line integrated velocity along each line of sight is given by

$$u = \left(x_{0ref} - x_{0shot}\right) * D(\lambda_{shot}) \tag{5.7}$$

where x_{0ref} and x_{0shot} are the centroid locations in pixel space and λ_{shot} is the spectrometer central wavelength during the shot. In some cases; however, the spectrometer central wavelength was changed slightly (>0.5nm) from the reference image position between shots. To account for this the reference data is shifted by

$$\Delta\text{px}_{\text{correction}} = \frac{\lambda_{shot} - \lambda_{cal}}{R\left(\lambda_{average}\right)} \tag{5.8}$$

where λ_{cal} is the spectrometer central wavelength for the calibration image, and $\lambda_{average}$ is the mean value of the two wavelengths. The error in the velocity is thus given by

$$\sigma_v = |u| \sqrt{\frac{\sigma_{x_0 ref}^2 + \sigma_{x_0 shot}^2 + \sigma_{\Delta px_{correction}}^2}{\left(x_{0ref} + \Delta px_{correction} - x_{0shot}\right)^2} + \left(\frac{\sigma_{D(\lambda)}}{D(\lambda)}\right)^2} \tag{5.9}$$

where $\sigma_{x_0 ref}$ and $\sigma_{x_0 shot}$ are the centroid errors returned from the curve fit to the reference and shot intensity profiles, respectively, $\sigma_{\Delta px_{correction}}$ is the error in the correction shift and $\sigma_{D(\lambda)}$ is the error in the Doppler resolution. For $\sigma_{\Delta px_{correction}}$ and $\sigma_{D(\lambda)}$ the error is propagated due to uncertainty in the spectrometer central wavelength $\sigma_\lambda = 0.005$nm using standard error propagation formulas.

For shots where the reference image was taken using a spectral lamp (typically H_α), the line integrated temperature can be computed as

$$T = \frac{mc^2}{k_B \lambda}\left(R(\lambda)^2\left(\sigma_{shot}^2 - \sigma_{ref}^2\right)\right) \tag{5.10}$$

where σ_{shot} and σ_{ref} are the standard deviation of the gaussian fits (in pixel space) to the shot and reference fiber images respectively. The error associated with the calculated temperature is

$$\sigma_T = |T| \sqrt{\left(\frac{2\sigma_{R(\lambda_{shot})}}{R(\lambda_{shot})}\right)^2 + \frac{4\left(\left(\sigma_{\sigma shot}\sigma_{shot}\right)^2 + \left(\sigma_{\sigma ref}\sigma_{ref}\right)^2\right)}{\left(\sigma_{shot}^2 - \sigma_{ref}^2\right)^2}} \tag{5.11}$$

where $\sigma_{\sigma shot}$ and $\sigma_{\sigma ref}$ are errors in the gaussian standard deviation returned from the curve fits and $\sigma_{R(\lambda_{shot})}$ is the error in the dispersion relation at the given wavelength due to uncertainty in the spectrometer central wavelength (propagated using standard error propagation formulas).

5.3.1 Hydrogen plasmas - H_α emission

The first spectral line measurements taken with the multichannel optical fiber bundle were of H_α emission at 656.3 nm. Because the reference image was taken using a spectral lamp, and because the primary source of line broadening of this emission line is Doppler broadening, line integrated velocities as well as temperatures can be calculated. Contribution due to stark broadening is ignored but can be estimated, where the Stark FWHM is given by $2.5 \times 10^{-14}\ \alpha_{1/2} n_e^{2/3}$ and $\alpha_{1/2}$

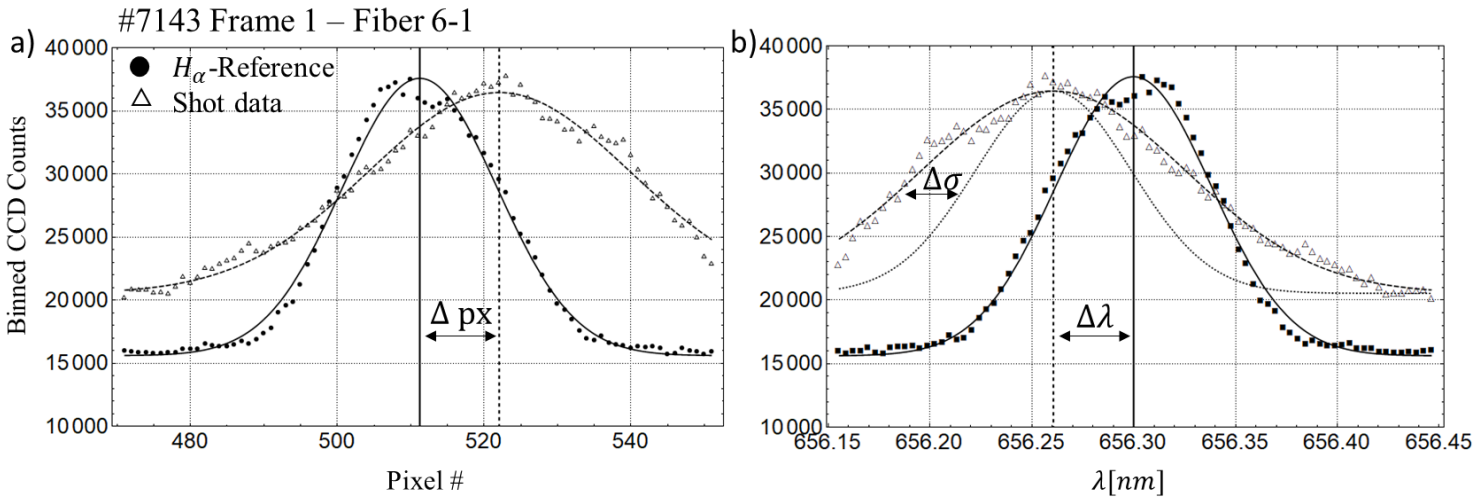


Figure 5.3. Comparison between shot data and the reference image for fiber 6-1 in pixel space (a) and wavelength space (b). Line integrated velocities are calculated from Δpx between the two centroids (related to $\Delta\lambda$ through the spectrometer dispersion relation) while temperatures are calculated from $\Delta\sigma$, based on the assumption that the standard deviation of the reference gaussian is representative of the instrument temperature. This instrument function (solid curve) is shifted and rescaled (dotted curve in panel (b)) for comparison with the shot data gaussian (dashed curve).

is tabulated. For our plasmas $\alpha_{1/2} \approx 0.00352$ and the stark contribution is not expected to exceed $\sim 5eV$. To measure axial and azimuthal velocities, the twelve fiber bundles were split into two measurement planes, a poloidal plane for measurement of axial velocities and a toroidal plane for measurement of azimuthal velocities (Figure 5.4). Based on this arrangement, a large transition from redshift to blue shift was expected as one moves from fiber bundle 1 on the gun side to fiber bundle 6 on the far side. In the azimuthal plane, rotation was expected to manifest as a transition between blue and red shifts with increasing or decreasing chord number for each bundle (depending on the direction of rotation). Because the exact orientation of the jet is not known, line integrated velocities are not adjusted to correct for the relative viewing angle of the chords (Table 5.1). Consequently, for jets with a uniform axial flow velocity aligned with the $\hat{z}$ axis, a negative gradient in the poloidal line integrated velocities is expected, with a velocity ratio between view chords given by

$$r_{v_{ij}} = \cos(\theta_i)\,/\cos(\theta_j) \tag{5.12}$$

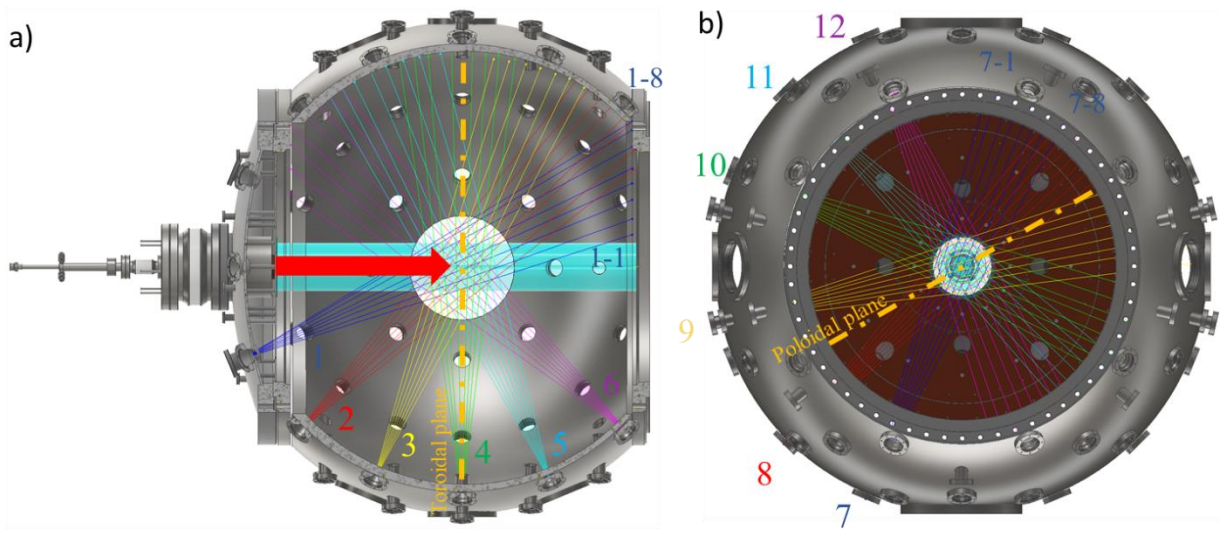


Figure 5.4. Optical fiber bundles were divided into two measurement planes and viewing chords were determined by back-shining a laser through the fibers. a) The poloidal measurement plane for measurement of axial velocities. b) Toroidal measurement plane for measuring azimuthal velocities.

where $\theta_{i/j}$ is the angle of chord i/j with respect to the $\hat{z}$ axis. Equation 5.12 further presumes that each poloidal line of sight crosses directly through the center of rotation such that azimuthal rotation does not contribute to the measured velocity. However, because there are small deviations between individual chords off the poloidal plane, the effects of azimuthal rotation will likely impact the line integrated velocities in the poloidal plane. This will be particularly important for jets that are not well aligned with the $\hat{z}$ axis where the impact of azimuthal rotation on the line integrated poloidal velocities may be significant.

Figure 5.5 shows the evolution of a typical plasma jet during this operation. Spectra were measured at $t = 43\mu s$ and $t = 58\mu s$ with an exposure time of $0.5\mu s$ and an intensifier gain of 50. The resulting line integrated velocities in the poloidal plane (figure 5.6) exhibit the predicted behavior of increasing blue shift when moving from fiber bundle 1 to fiber bundle 6 for both frames. However, a substantial negative skew ($\sim -10 km/s$) to the data is seen in both frames such that the expected large redshifts for fiber bundles 1-3 are not observed. Because the spectrometer was not adjusted between the reference image and the shot, it is unlikely that the source of skew is due to a systematic shift in the fiber positions on the CCD. Instead, based upon the orientation of the jet which appears to bow slightly downwards and towards the far side of the

chamber, it is possible that the view chords in the poloidal plane intercept the top half of the jet and are affected by strong azimuthal rotation in the counter-clockwise direction leading to a systematic blue shift in the observed line integrated velocities. This explanation; however, does not account for a similar skew observed for fiber bundles in the toroidal plane (figure 5.7). Other possible explanations for this skew include large radial outflows from the central plasma column or bulk oscillation of the column towards the optical fiber bundles. The expected negative gradient in the line integrated poloidal velocities is also observed for most fiber bundles in the first frame, and several bundles in the second frame; however, the magnitude of the gradient for several bundles (i.e. bundle 1 in frame 1 and bundles 2 and 5 in frame 2) is larger than predicted from equation 5.12 suggesting other effects such as rotation or axial sheared flow are contributing to this profile.

In the toroidal measurement plane there are indications of significantly sheared azimuthal flows for fiber bundles 7 and 8, with an increase in the magnitude of these flows between frames. Because the jet is bowed downwards at these times, it is possible that fiber bundles 9-12 do not intercept the central plasma column and thus do not present similar indications of strong flows. Even accounting for the negative skew, the line integrated velocity profiles for bundles 7 and 8 do not appear to reflect what would be expected for coherent azimuthal rotation (i.e. a diagonal

Table 5.1

Angle between view chords in the poloidal plane and the $\hat{z}$ axis*

Bundle	Chord 1	Chord 2	Chord 3	Chord 4	Chord 5	Chord 6	Chord 7	Chord 8
1	27°	28°	30°	32°	33°	35°	37°	38°
2	43°	43°	47°	48°	50°	52°	53°	55°
3	65°	67°	68°	70°	72°	74°	76°	-----
4	84°	86°	87°	89°	91°	94°	96°	97°
5	103°	106°	108°	110°	112°	114°	115°	117°
6	126°	128°	130°	132°	133°	135°	137°	140°

*Angles are approximate and calculated based on camera images of the laser backlit fibers

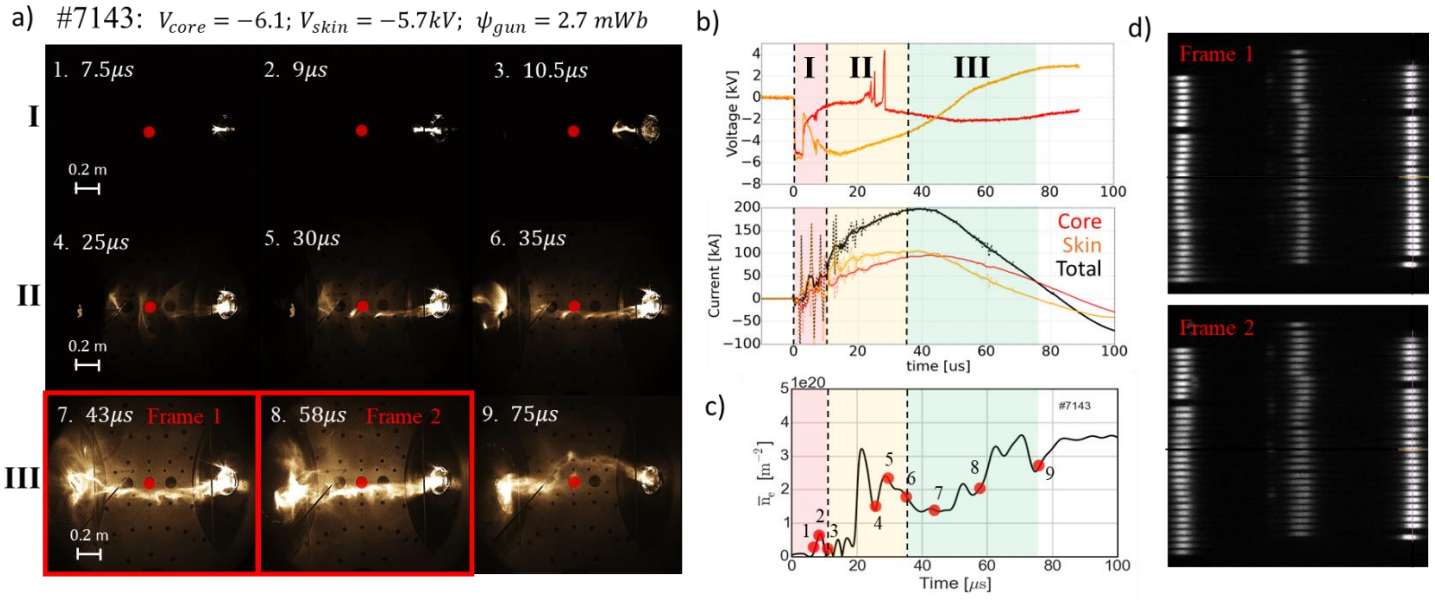


Figure 5.5. Evolution of shot #7143. a) High speed camera images taken with the Kirana camera at 2Mfps. The red dot indicates the path of the interferometer. Panels 7 and 8 correspond to the timing of the first and second frames taken with the PI-MAX3 ICCD camera. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. d) ICCD images of the optical fiber bundle taken using the PI-MAX3 ICCD camera at 43μs and 58μs with an exposure time of 0.5μs.

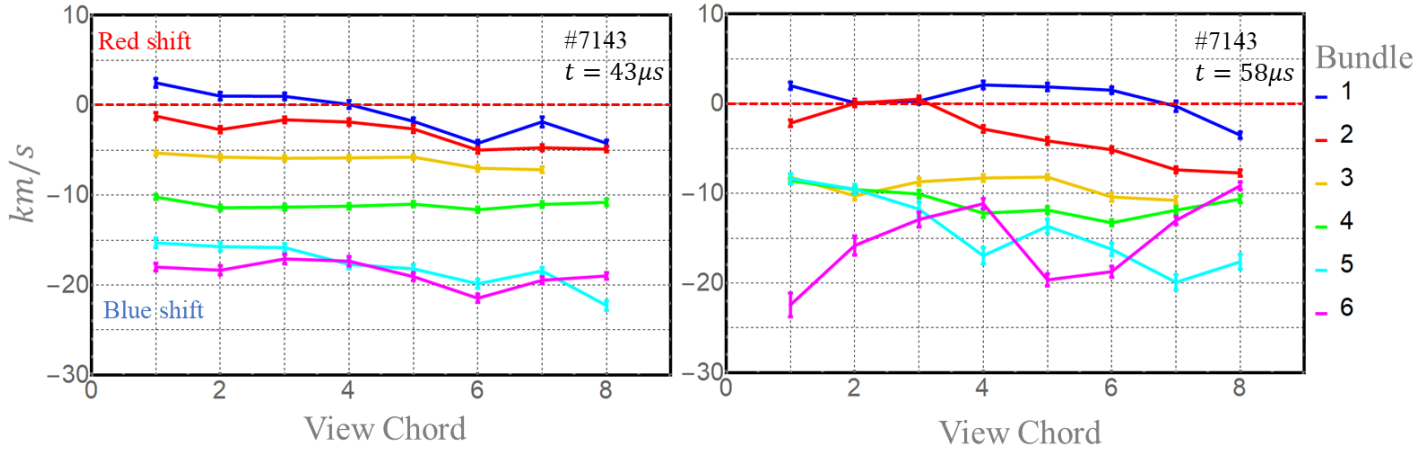


Figure 5.6. Line integrated velocities in the poloidal plane for shot #7143 using H_α emission lines. Increasing blue shift from fiber bundle 1 to fiber bundle 6 agrees with the predicted behavior; however, a negative ($\sim -10 km/s$) skew to the data is observed. Possible sources for the skewing of the data set include large radial outflows, bulk oscillation of the plasma column, or contributions from an azimuthal velocity.

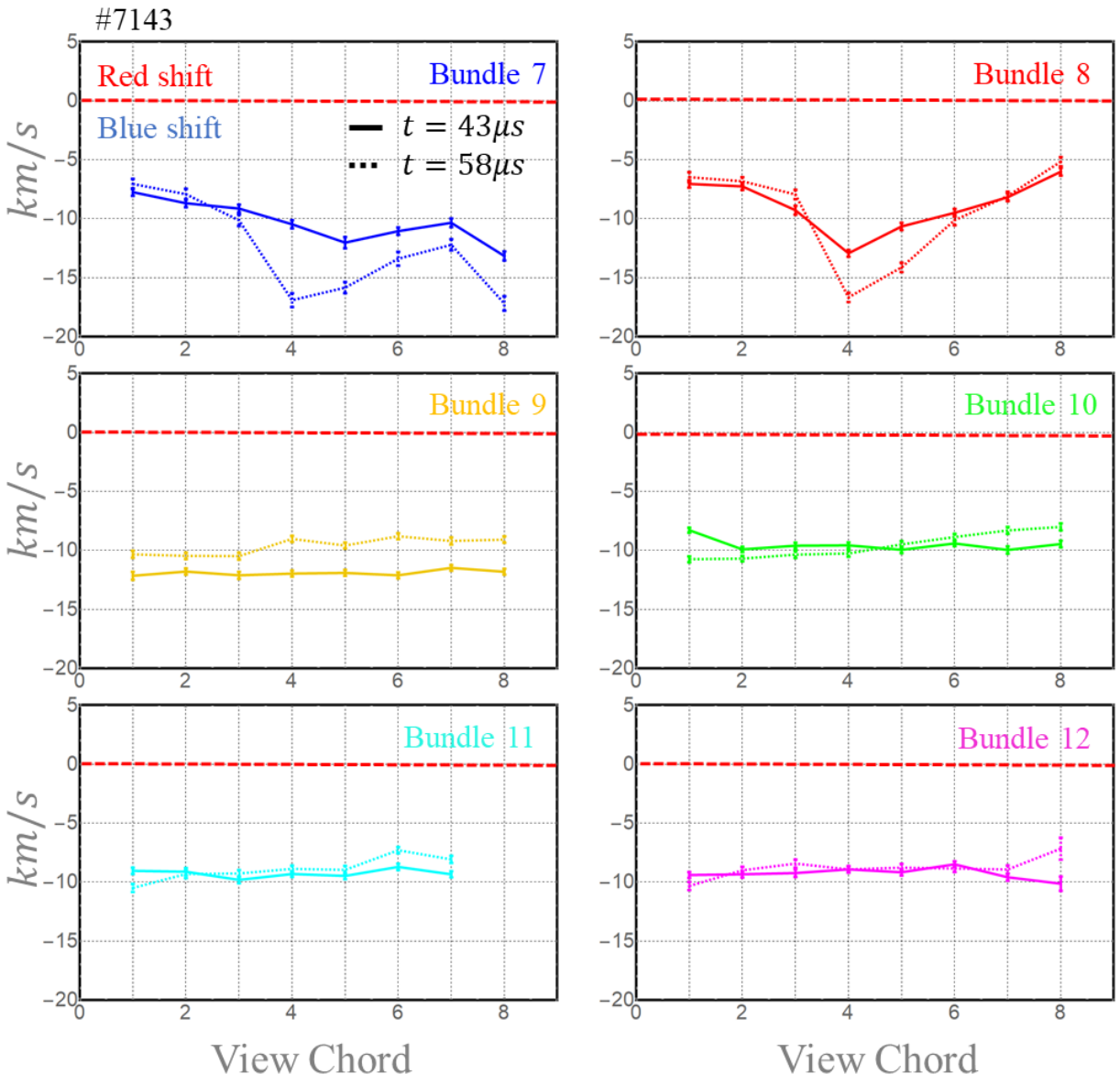


Figure 5.7. Line integrated velocities in the toroidal plane for shot #7143 using H_α emission lines. Indications of significant flow shear are observed for fiber bundles 7 and 8 but are not observed for the other fiber bundles which likely do not intercept the central plasma column. A negative skew to the data set of $\sim -10 km/s$ may be due to large radial outflows or bulk oscillation of the plasma column

line increasing/decreasing with fiber number for counter- clockwise/clockwise rotation). Instead the profiles exhibit an increasing blue shift up to fibers 4-5, where the line integrated flows then begin to decrease in magnitude. Fiber bundle 7 has a second large decrease in velocity for chord 8 that is not observed for fiber bundle 8. At $t = 59\mu s$ the line integrated velocities for fiber bundles

9-11 do exhibit a small positive slope which may correspond to azimuthal rotation in the counter-clockwise direction.

Line integrated temperature measurements in the poloidal plane (figure 5.8) indicate significant heating from $\sim$7 eV at $t = 43\mu s$ to $\sim$23eV at $t = 58\mu s$ despite a decrease in total jet current of $\sim 90kA$. Based on the initial bowing of the plasma column, this temperature spike may reflect the fact that the view chords do not intercept the central plasma during the first frame. Similar temperatures observed in the toroidal plane at $t = 43\mu s$ (figure 5.9) however, suggest that the heating is not simply the result of changes in jet orientation. Furthermore, the observed heating supports previous temperature estimates obtained by assuming Spitzer resistivity (Figure 4.17). Gradients in the poloidal line integrated temperature profiles for individual fiber bundles however, may reflect whether the individual view chords intercept the central plasma column. This conjecture appears to be corroborated when examining the temperature profiles in the toroidal plane where more significant heating is observed for fiber bundles 7 and 8 than bundles 9-12. Interestingly, bundles 7 and 8 exhibit a double peaked temperature profile with peak temperatures of $\sim$23eV in the wings (in agreement with measurements in the poloidal plane) and a minimum temperature of $\sim$10eV at the center of the profile. For fiber bundles 9-12 which do not appear to cross the central plasma column, subtle heating from $\sim$5 eV to $\sim$10 eV is observed.

5.3.2 Hydrogen plasmas – CII impurity emission

Despite precautions taken to keep the vacuum system clean, impurities present within the plasma radiate line emission which can be used for IDS. For hydrogen plasmas, the brightest impurity lines within the operational wavelength range of the spectrometer and fiber bundle was found to be a CII doublet at 723.132 nm and 723.642 nm with relative intensities of 800 and 1000 respectively [91]. Unfortunately, for these spectral lines no suitable calibration line could be found using available spectral lamps so reference images were taken using a long exposure (2 ms) of a plasma discharge with the assumption that the time integrated velocities along each line of sight average to zero (Figure 5.2b). This reference image thus permits calculation of line integrated velocities but precludes temperature measurements since the reference is itself Doppler broadened.

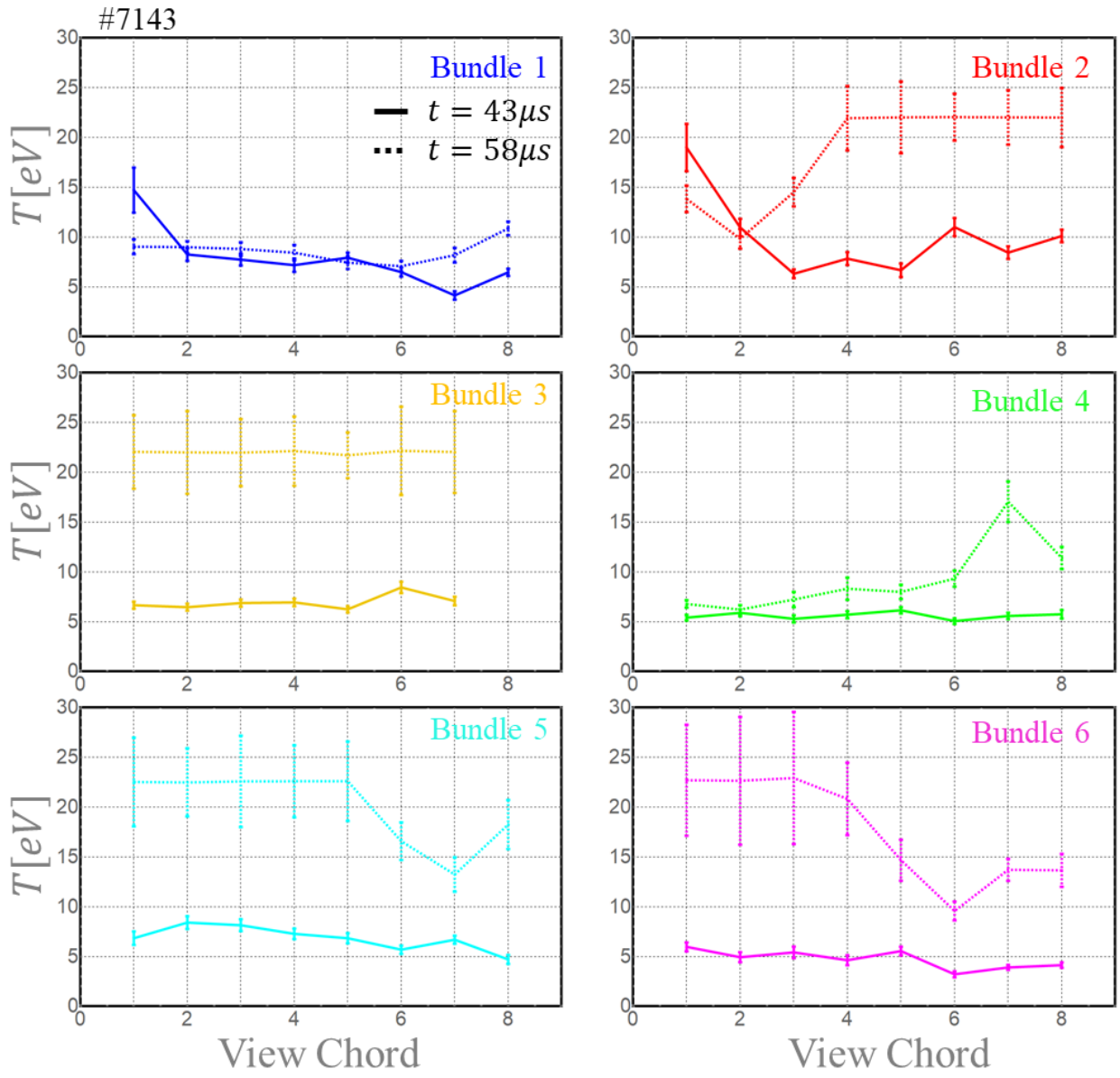


Figure 5.8. Line integrated temperatures in the poloidal plane for shot #7143 using H_α emission lines. Significant heating is observed between $t = 43\mu s$ and $t = 58\mu s$. Sharp gradients in the temperature profile likely reflect whether the lines of sight are intercepting the central plasma column where temperatures are expected to be largest.

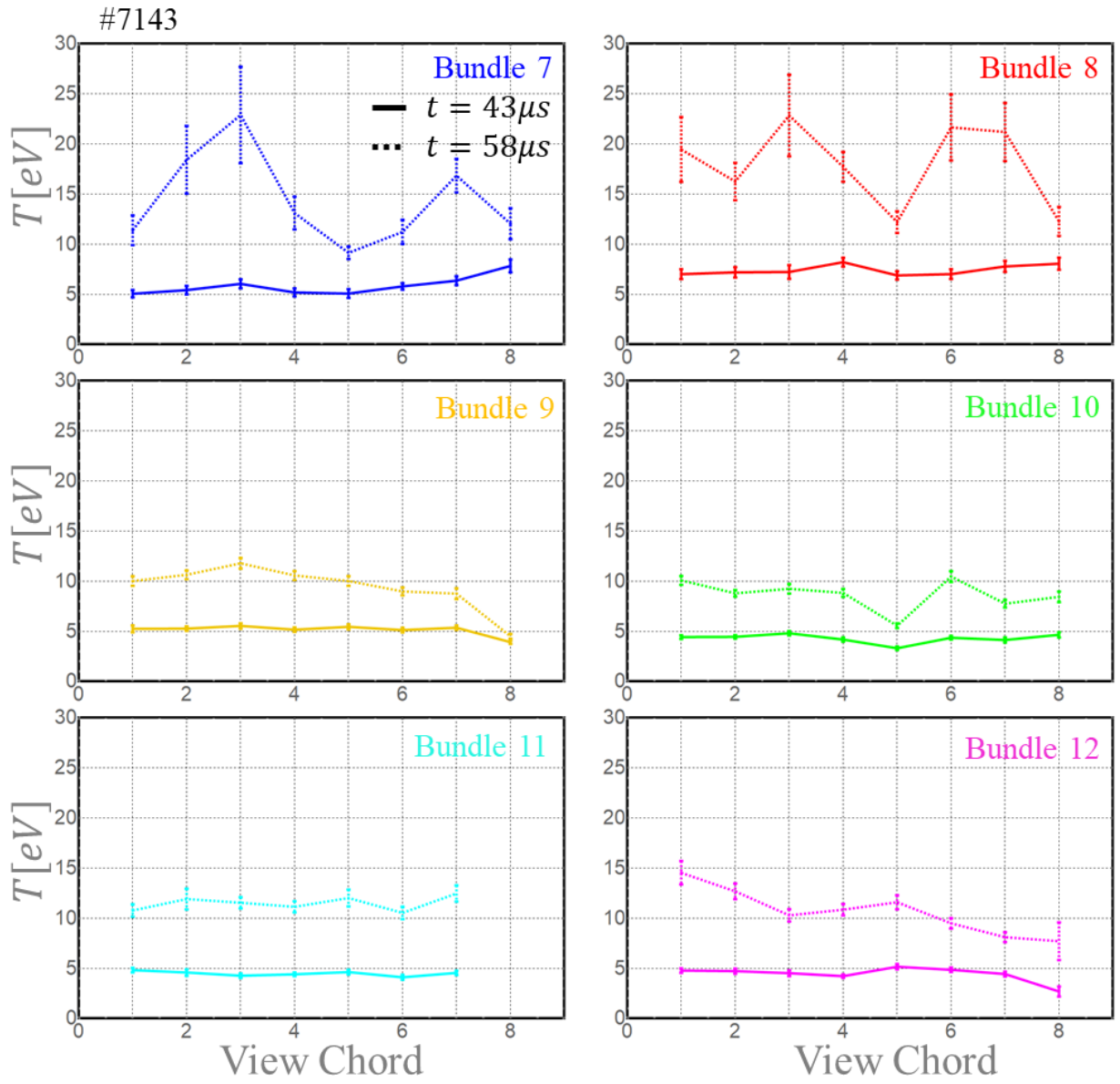


Figure 5.9. Line integrated temperatures in the toroidal plane for shot #7143 using H_α emission lines. The most pronounced heating is observed for fiber bundles 7 and 8 which exhibit a double peaked temperature profile. Less significant heating is observed for the remaining bundles which likely do not intercept the central plasma column.

The full evolution of a typical discharge during this operation is presented in figure 4.16 while frames corresponding to the measurement of CII line emission are shown in figure 5.10. Because the CII lines are weaker than Hα, the exposure of the ICCD was doubled to $1\mu s$ with an intensifier gain of 100 to obtain a satisfactory signal while still having sufficient time resolution. In the first frame at $t = 43\mu s$, the central column of the jet appears to be bowed downwards towards the far side of the chamber (where the fibers are mounted) and fiber bundles 3-5 in the poloidal plane present a very weak signal since they do not intercept the jet. The same is true for fiber bundles 9-12 in the toroidal plane where only the edge fibers appear to intercept the jet. In the second frame at $t = 59\mu s$ the jet has straightened and all fiber bundles show an increased intensity.

Ignoring channels that do not intercept the central plasma column, line integrated velocities in the poloidal plane again obey the predicted behavior; with a strong transition from red-shift to blue-shift with increasing bundle number (Figure 5.11). In the first frame, strong red shifts of $\sim$40km/s are observed for fiber bundle 1 with comparably large blue shifts observed for fiber bundle 6. A significant negative gradient for fiber bundle 6 exceeds what would be predicted from equation 5.12 while bundles 1 and 2 do not appear to have any coherent negative gradient. Instead, a small dip in velocity for chords 1-3 of fiber bundle 1 may reflect deceleration of the flow due to impact and pileup at the end of the chamber. This feature is even more pronounced in the second frame, where line integrated velocities for bundle 1 increase from $\sim$25 km/s to $\sim$50 km/s between view chords 1 and 6 with a similar increase observed for fiber bundle 2. Fiber bundle 6 however, shows a marked decrease in velocity in frame 2 with no substantial negative gradient, suggesting that the axial flows within the jet could be decelerating as the current drops and the driving $\vec{j} \times \vec{B}$ force is reduced. Another possibility is that the change in orientation of the jet is altering the contribution of the azimuthal velocity to the total line integrated velocity, resulting in the different profiles between frames. This idea is supported by a large offset observed in frame 2 for fiber bundle 4 where, for jets perfectly aligned with the $\hat{z}$ axis with a uniform axial flow velocity, the velocity profile is expected to cross from positive to negative between chords 4 and 5.

In the toroidal plane, measurements are indicative of strong azimuthal flows (Figure 5.12). Focusing on bundles 7-8 in the first frame, a velocity profile transitioning from blue to red shift with increasing view chord number is consistent with azimuthal rotation in the counter clockwise direction. For fiber bundle 7, this profile appears to be shifted to the left compared to bundle 8,

with a transition from $-3.6\ km/s$ to 6 km/s between fibers 1 and 4 and decreasing again, presumably at the edge of the jet. For fiber bundle 8, this transition is well centered, going from -8.9 km/s to 8.1 km/s between fibers 2 and 5. This rotation profile is even more pronounced in the second frame with a transition from $-10.7\ km/s$ to 13.5 km/s for fiber bundle 7 and $-8.1\ km/s$ to 20.9 km/s for fiber bundle 8. In the second frame, when the jet is well centered, fiber bundle 12 also exhibits a similar transition from $-8.5\ km/s$ to 15.9 km/s while bundle 11 seems to show the right-hand side of the same profile, increasing from $-1.2\ km/s$ to 9.7 km/s between fibers 1 and 5 before decreasing again, presumably at the edge of the jet. In Frame 2, fiber bundle 9 also displays a similar profile but is offset from 0 by $\sim 10 km/s$, possibly due to the contribution of bulk oscillation away from the far side of the chamber. Such an oscillation would be expected to effect bundles 9 and 10 more substantially than bundles 7, 8, 11, and 12 which view from the top and bottom of the chamber.

Compared to observations taken using H_α emission, significantly larger line integrated velocities, the lack of a coherent velocity offset, and indications of azimuthal rotation all suggest that the carbon impurity ions obey distinctly different dynamics than the neutral hydrogen. The lack of any apparent offset suggest that the impurity ions are well confined by the magnetic fields while the neutral particles are free to leave the plasma perpendicular to the magnetic field and radiate outward towards the chamber walls. Indications of counter-clockwise rotation also agree with the predicted $\vec{E} \times \vec{B}$ direction of the gun where the radial electric field points inwards towards the negatively biased middle and inner electrodes, while the bias magnetic field points out of the gun.

5.3.3 Nitrogen plasmas – NII emission

The formation of long, stable jets was also observed when using NII as the working gas. A strong NII emission line within the operational range of the monochromator and fiber bundle was found at $648.205\ nm$. Unfortunately, this spectral line was too weak to be resolved using a calibration lamp so instead a long exposure ($2ms$) of the plasma discharge was used as the reference image (figure 5.2c). This reference permits calculation of line integrated velocities but precludes temperature measurements since the reference is itself Doppler broadened. Contributions to the emission line due to nearby NI emission can be seen in reference and shot images but are discounted by reducing the horizontal extent of the binning rectangels.

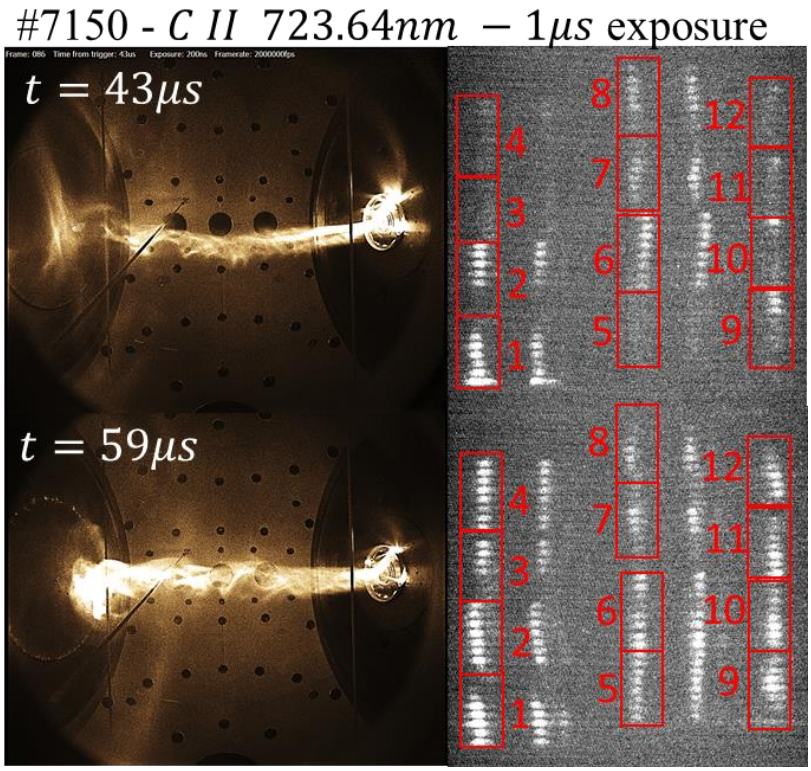


Figure 5.10. Kirana camera images of a typical jet at $43\mu s$ and $59\mu s$ with the corresponding ICCD images of the optical fiber bundle measuring CII impurity emission at 723.64nm (red boxed). A second impurity line corresponding to CII at 723.13nm is also observed but was not used as it was slightly weaker. The full evolution of this shot, including gun current, voltage, line integrated electron density measurements and stability space analysis are presented in figure 4.16.

Figure 5.13 shows the evolution of a typical plasma jet during this operation. Spectra were measured at $t = 43\mu s$ and $t = 58\mu s$ with an exposure time of $1\mu s$ and an intensifier gain of 100. The resulting line integrated velocities in the poloidal plane are significantly slower than those observed for CII and are more comparable to the magnitudes observed for H_α. The expected transition from red shift to blue shift between bundles 1 and 6 is observed in the first frame, but is much noisier than for CII and H_α emission. Interestingly, the profile for bundle 6 at $t = 43\mu s$ is similar to the profile observed for shot #7143 at $t = 58\mu s$ using H_α emission, with a decrease in velocity up to chord 4, a sharp increase at chord 5 and a decrease again up to chord 8. Such a profile is not easily explained by a jet with a uniform axial flow velocity and may represent contributions due to azimuthal rotation or the presence of strong axially sheared flows. In the

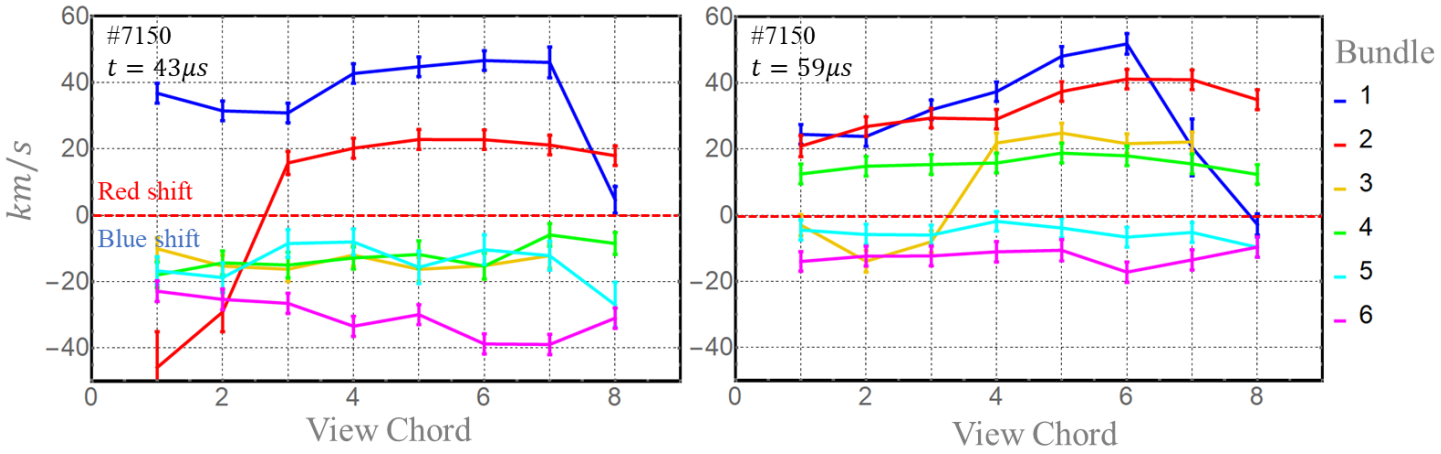


Figure 5.11. Line integrated velocities in the poloidal plane for shot #7150 using CII emission lines. Transitions from large red shift to blue shift with increasing bundle number follows the predicted behavior. Steep positive gradients for bundles 1 and 2 are observed at $t = 59\mu s$ possibly indicating deceleration due to pileup at the end wall. A large decrease in velocity between frames is also observed for bundle 6.

second frame, bundles 1-5 are blue shifted by $\gtrsim 5$ km/s and do not exhibit the same increasing blue shift with increased bundle number. For bundles 1 and 2, this may be the result of the plasma column colliding with the end wall and propagating backwards while for bundles 3-5, a subtler shift may be the result of bulk motion of the central plasma column away from the fibers.

In the toroidal plane, velocity profiles for each bundle are similar between frames but also exhibit a corresponding blue shift. This corroborates the possibility that bulk oscillation of the central plasma column may responsible and agrees with high speed video which shows the jet oscillating slightly towards the fiber bundles. The magnitude of the line integrated velocities in the toroidal plane are substantially weaker than those observed using CII emission for hydrogen jets. Gradients in the velocity profiles are indicative of sheared flows; however, the profile shape is not coherent across fiber bundles. Bundles 7 and 12 for example have a positive gradient with a maximum Δv of $\sim$5km/s while bundles 8 and 10 have a negative gradient overall with a small positive gradient between chords 3-4 and 4-5 respectively. This nonuniformity in velocity profiles may result from the fact that this jet was very narrow and as such many of the view chords do not intercept the central plasma column and are instead picking up turbulent flows outside of the jet.

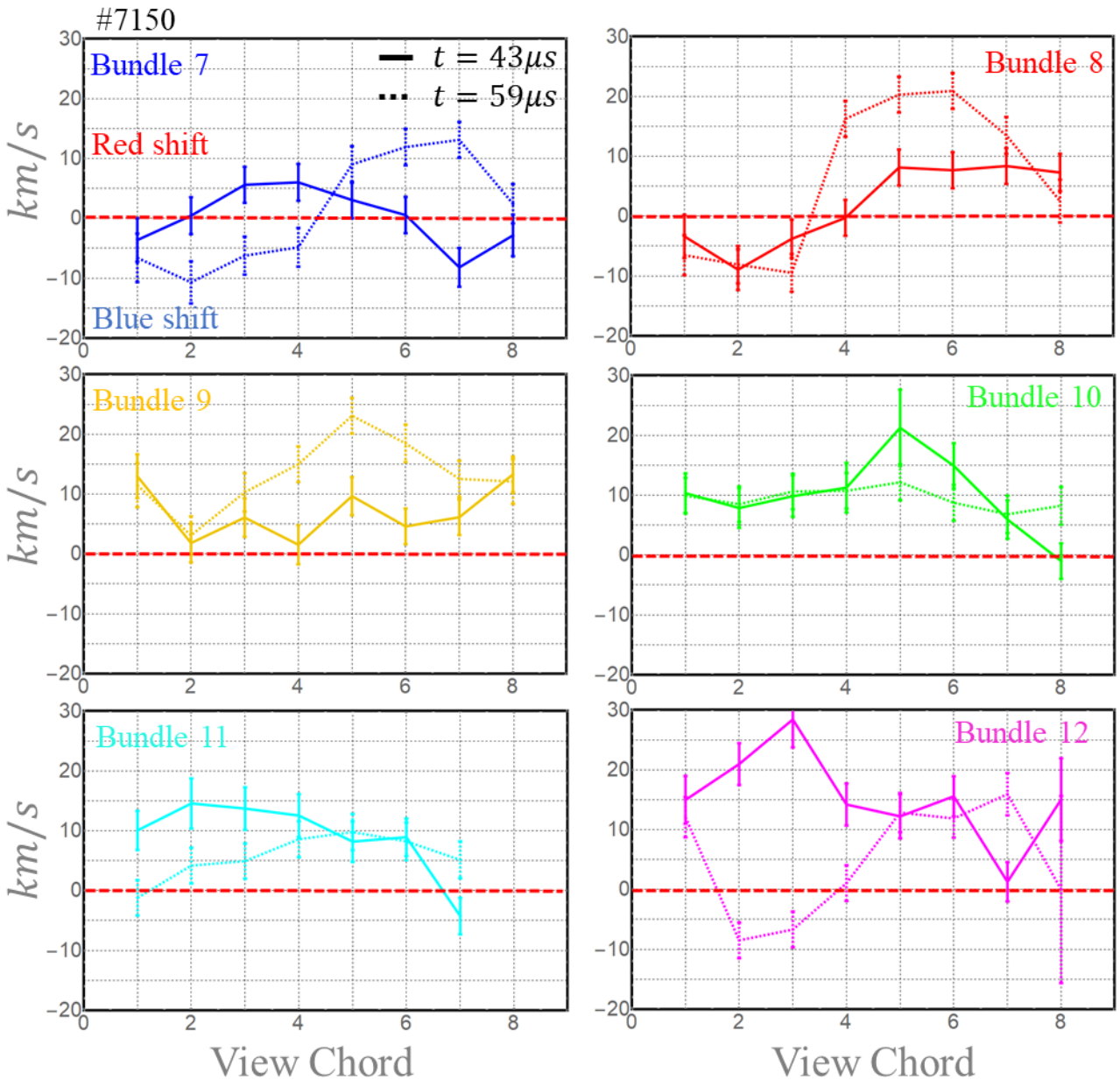


Figure 5.12. Line integrated velocities in the toroidal plane for shot #7150 using CII emission lines. At $t = 43\mu s$ the velocity profiles for bundles 7 and 8 are indicative of azimuthal rotation in the counter-clockwise direction based on a transition from blue shift to red shift with increasing view chord number. Similar profiles are observed for all bundles except bundle 10 at $t = 59\mu s$. This rotation direction is consistent with the predicted $\vec{E} \times \vec{B}$ direction of the LabJet gun.

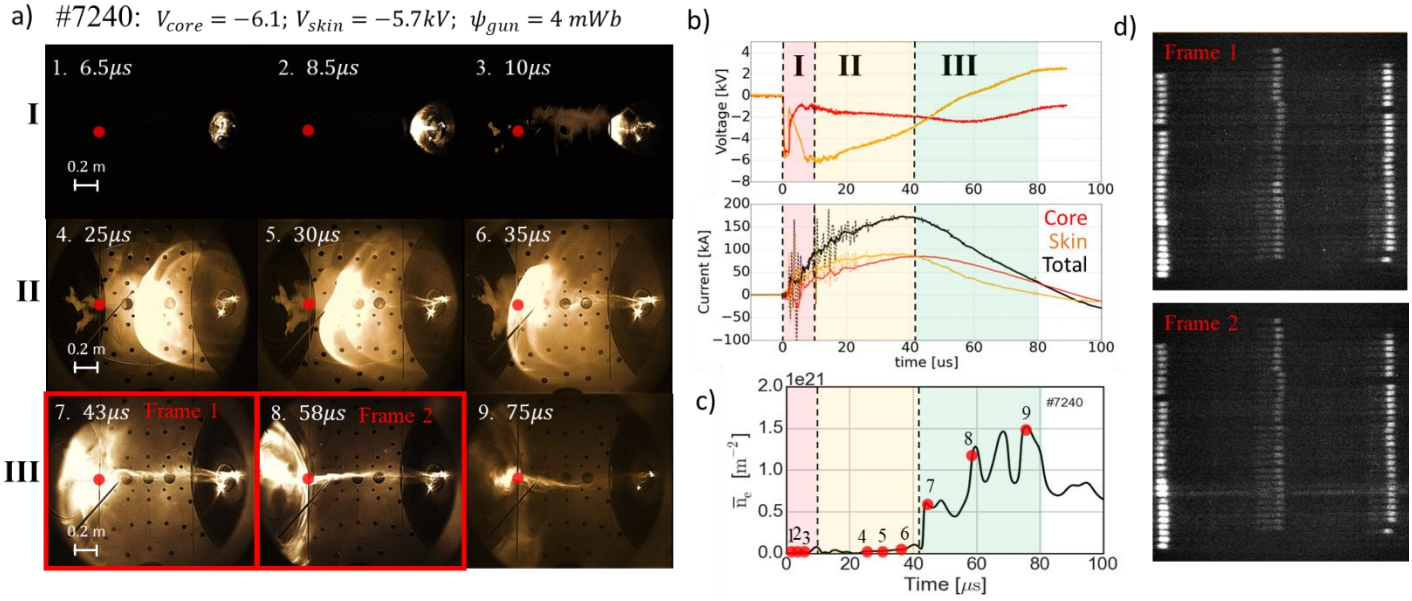


Figure 5.13 Evolution of shot #7240 with NII as the working gas. a) High speed camera images taken with the Kirana camera at 2Mfps. The red dot indicates the path of the interferometer. Panels 7 and 8 correspond to the timing of the first and second frames taken with the PI-MAX3 ICCD camera. b) Gun voltage and current waveforms measured using Tektronix HV probes and calibrated, passively integrated Rogowski coils. c) Evolution of the line integrated electron density. d) ICCD images of the optical fiber bundle taken using the PI-MAX3 ICCD camera at 43μs and 58μs with an exposure time of 1μs at a central wavelength of 648.2nm.

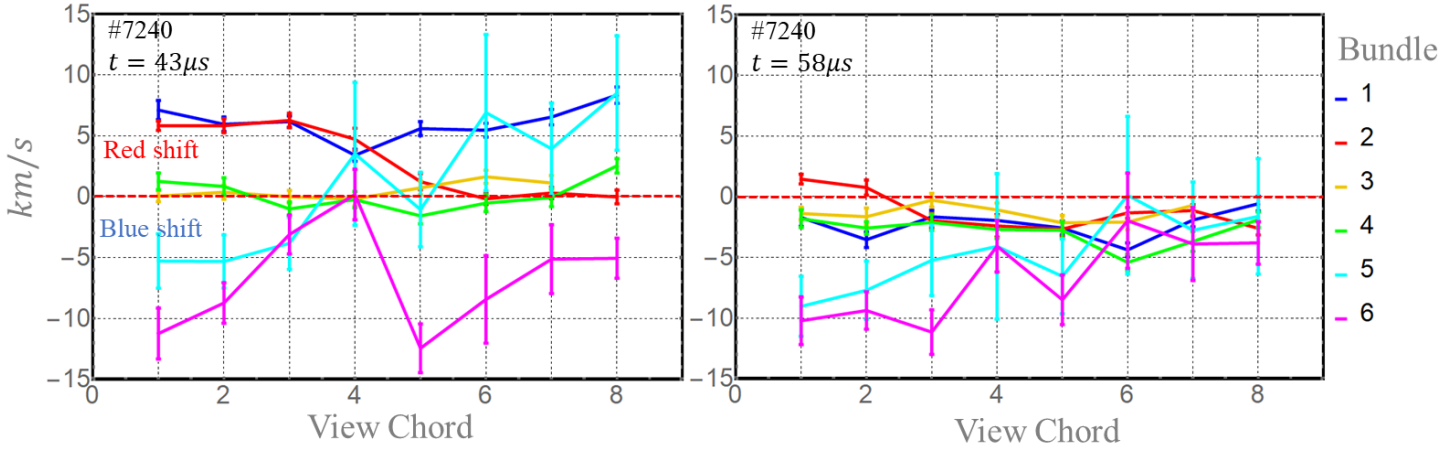


Figure 5.14. Line integrated velocities in the poloidal plane for shot #7240 using NII emission lines. Transitions from red shift to blue shift with increasing bundle number at $t = 43\mu s$ follows the predicted behavior. At $t = 58\mu s$ bundles 1-5 are blue shifted by $\gtrsim$ 5km/s possibly due to bulk oscillation of the central column and/or deceleration of the jet due to impact and reflection off the end wall.

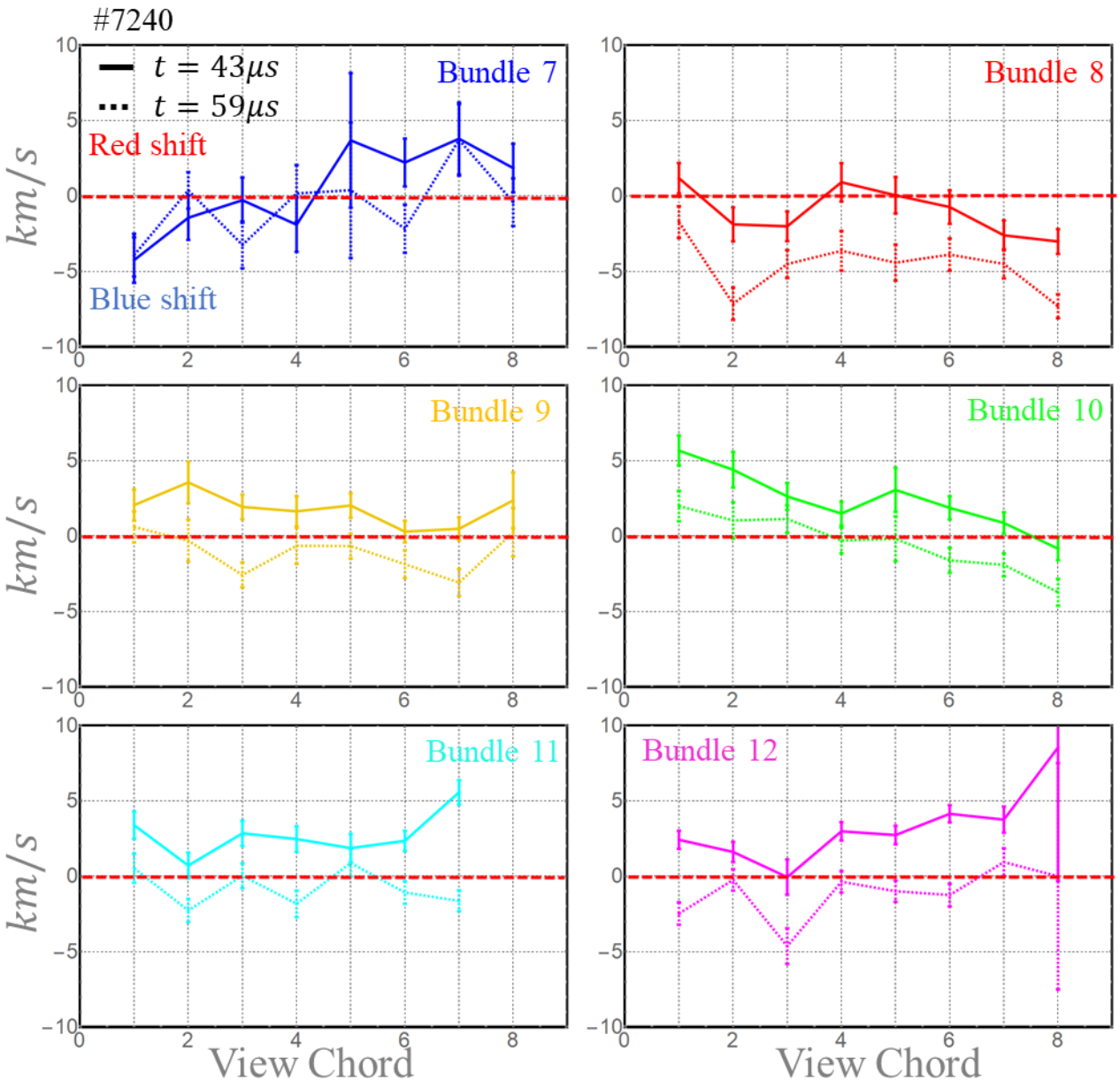


Figure 5.15. Line integrated velocities in the toroidal plane for shot #7240 using NII emission lines. A coherent blue shift of ~5 km/s is seen between frames, indicating bulk motion of the central plasma column. Gradients in the profile are indicative of sheared flows; however, the shape of the profiles are not coherent. Because this jet was quite narrow, this could reflect the possibility that most view chords are not intercepting the central plasma column and are instead picking up turbulent motion outside of the jet.

5.4 Estimating local plasma velocities

5.4.1 Double shell model

To provide an estimate of local plasma velocities, a simple helical shear flow model is fitted to the spectroscopic Doppler data. The model (figure 5.16) consists of a jet with two concentric cylindrical shells of equal thickness, a core region $0 \leq r \leq a/2$ and a skin region $a/2 \leq r \leq a$. The azimuthal velocity v_ϕ in each shell is specified independently. The approximate jet center position and diameter in the azimuthal measurement plane is determined by a gaussian fits to the spectral intensity as a function of view chord (figure 5.17 and figure 5.18). The model includes variables Δx, Δy, and Δa to adjust for uncertainties in the jet center and radius respectively. The axial flow velocity profile within the model is defined as

$$\vec{v}_{ax}(r) = v_0 \exp\left(\frac{r^4}{r^4 - a^4}\right)\hat{e}_{jet} \tag{5.13}$$

where v_0 is the maximum velocity on axis, a is the jet radius, $\hat{e}_{jet}$is a unit vector pointing from the center of the gun to the center of the jet in the azimuthal measurement plane, and r is the perpendicular distance from a line directed along $\hat{e}_{jet}$. The remaining model parameters, v_{xbulk} and v_{ybulk} capture any bulk motion of the plasma jet due to gyration and/or radial outflow. Assuming a uniform emissivity inside the plasma and zero emissivity outside, line integrated velocities through the model are calculated as

$$\bar{v}_i = \frac{\int_0^{L_i} \vec{v} \cdot d\vec{l}_i}{L_{pi}} \tag{5.14}$$

where L_i is the length of the *i*th line of sight and L_{pi} is the path length of the line of sight through the plasma model.

Figures 5.19 – 5.22 present the results of this analysis for shot #7150 while tables 5.2 – 5.5 list the corresponding model fit parameters. Despite the simplicity of the model, it provides remarkably good fits to the line integrated experimental data. At $t = 43\mu s$ model fits in the poloidal plane (figure 5.19, table 5.2) suggest strong axial flows on the order of 62 km/s with azimuthal flows of ~27km/s in the core and ~2km/s in the skin. Bulk velocities for the model fits agree with the predicted bulk velocities of $v_{xbulk} = -5.3\ km/s$ and $v_{ybulk} = -4.6\ km/s$ based on the distance between approximate jet center positions between frames. At the same time, fit

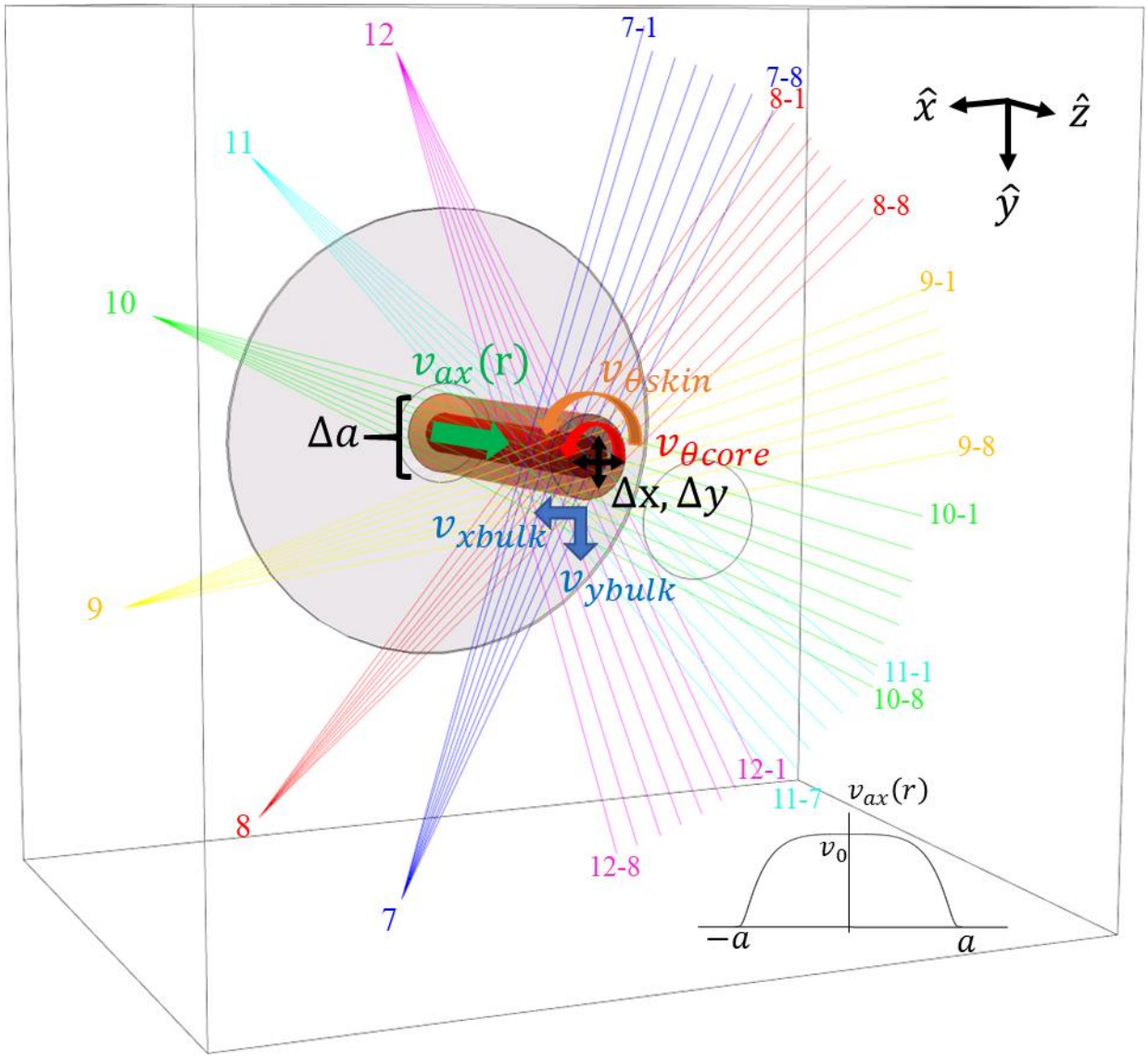


Figure 5.16. Simple spectroscopy model which consists of a jet with two concentric cylindrical shells of equal thickness. The approximate center and radius of the jet in the toroidal midplane is calculated using gaussian fits to the spectral intensity as a function of view chord (Figures 5.17 and 5.18). The model has 8 free parameters which can be adjusted independently. The axial velocity profile was selected to be relatively flat in the core, going smoothly to zero at the jet radius.

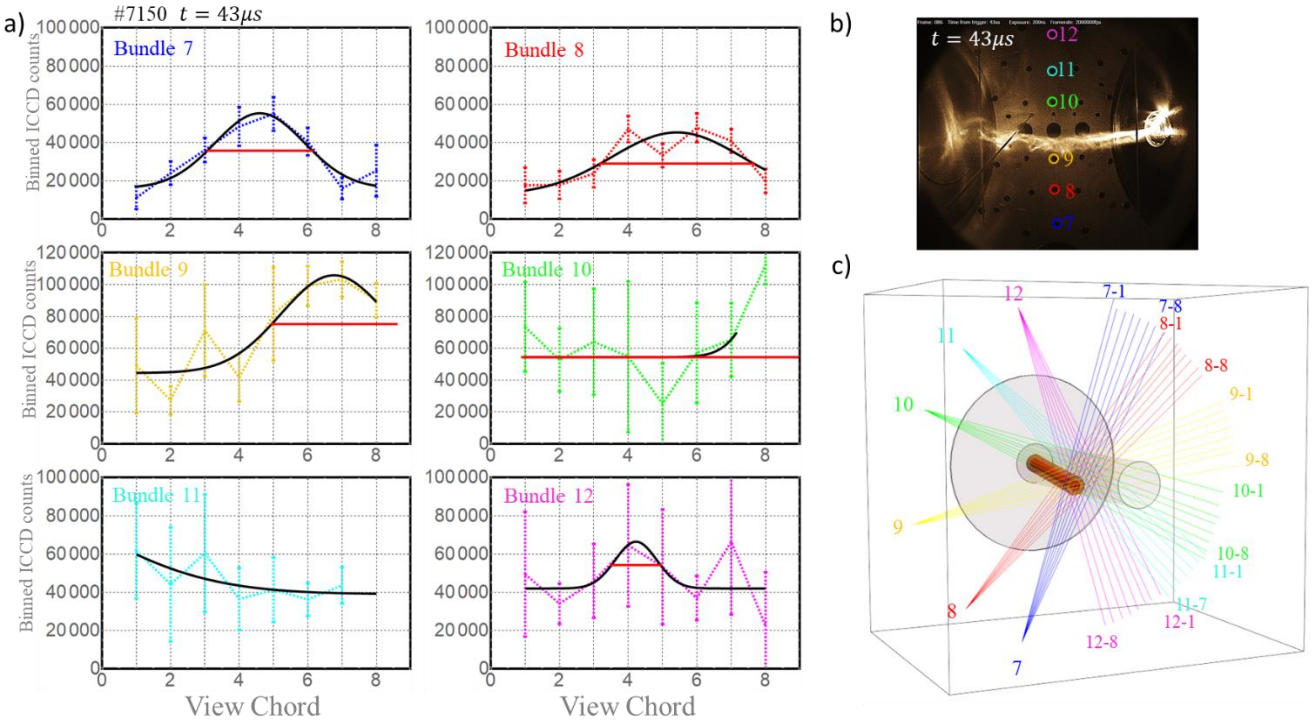


Figure 5.17. Calculation of the approximate jet center and radius for shot #7150 at $t = 43\mu s$. a) The centroid of gaussian fits to the total intensity (excluding bundles 10-12) allows for calculation of the approximate jet center position by minimizing the sum of the distances between a center point and the two view chords nearest to the centroid for each bundle. The approximate jet radius is calculated based on the average full width half max of the gaussian fits (excluding bundles 10-12) multiplied by the average spacing between view chords in the midplane ($\sim 2cm$). b) High speed camera image of the jet with fiber bundle viewports marked. c) Reproduction of jet model based on calculated center position and jet radius. The clear cylinder is a projection of the middle electrode while the grey circle marks the extent of the outer electrode.

parameters in the toroidal plane (Figure 5.20 and Table 5.3) for bundles 7-9 agree well, suggesting azimuthal velocities of ~ 15 km/s in the core and ~ 3 km/s in the skin. Bundles 10-12 which appear to overshoot the central plasma column require substantially different fit parameters for the location and radius of the column, reflecting an inherent weakness of the model in that it does not account for flows outside of the central jet. Still, a comparison of the average azimuthal velocity fit parameters in the poloidal and toroidal planes shows that they agree within the large standard deviation introduced primarily by bundles 1 and 12. Even if bundles 1 and 12 are ignored, the azimuthal velocity fit parameters in the poloidal plane agree well with the fit parameters in the toroidal plane.

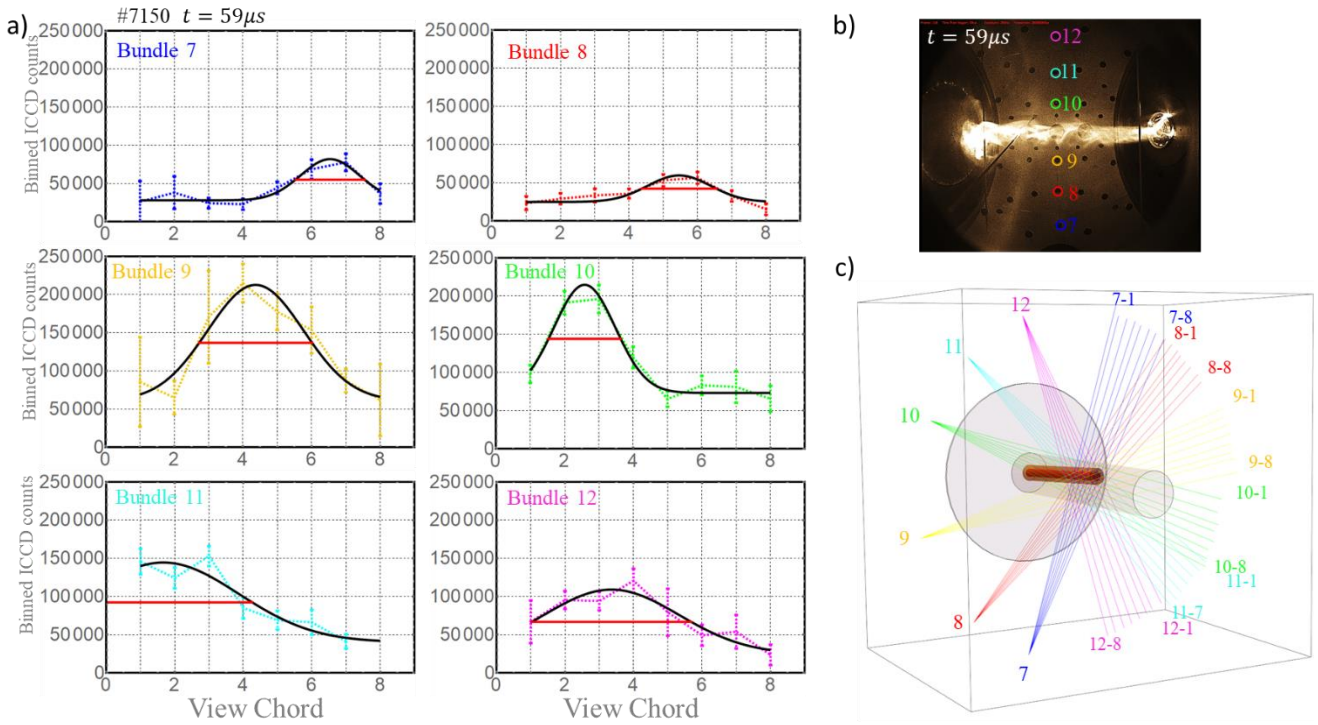


Figure 5.18. Calculation of the approximate jet center and radius for shot #7150 at $t = 59\mu s$. a) The centroid of the gaussian fits allows for calculation of the approximate jet center position by minimizing the sum of the distances between a center point and the two view chords nearest to the centroid for each bundle. The approximate jet radius is calculated based on the average full width half max of the gaussian fits multiplied by the average spacing between view chords in the midplane ($\sim 2cm$). b) High speed camera image of the jet with fiber bundle viewports marked. c) Reproduction of jet model based on calculated center position and jet radius. The clear cylinder is a projection of the middle electrode while the grey circle marks the extent of the outer electrode.

In the second frame, model fits to the line integrated velocity in the poloidal plane (Figure 5.21 and Table 5.4) suggest a deceleration of the axial velocity to ~ 54 km/s with a modest increase in azimuthal velocity in the core and skin to 32 km/s and 9 km/s respectively. The axial velocity fit parameter decreases with increasing bundle number suggesting that axial flows along the length of the jet may be decreasing as the driving current ramps down. In the toroidal plane (Figure 5.22 and Table 5.5), model fit parameters are in better agreement between bundles than in the first frame suggesting that all bundles intercept the central plasma column. Average azimuthal velocity fit parameters in the toroidal plane also indicate a modest increase in the skin azimuthal velocity to 9 km/s while the azimuthal velocity in the core remains relatively unchanged at 23 km/s. Comparison

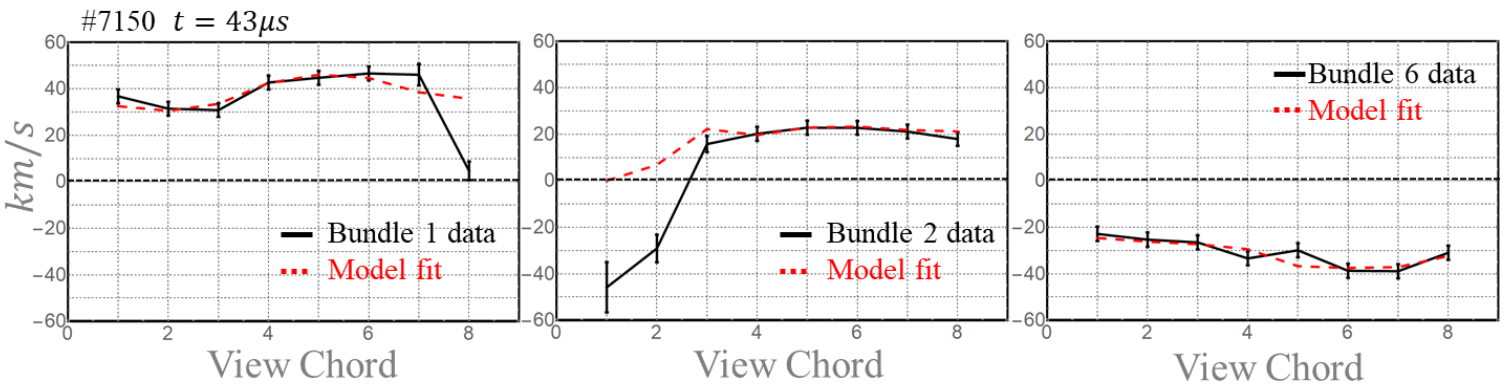


Figure 5.19. Double shell model fits to the line integrated velocities in the poloidal plane for shot #7150 at $t = 43\mu s$. Bundles 3-5 have been ignored due to low signal to noise for the spectral intensities.

Table 5.2
Model fit parameters for shot #7150 in the poloidal plane at $t = 43\mu s$*

Bundle	$\Delta x\ [m]$	$\Delta y\ [m]$	$\Delta a\ [m]$	$v_{\theta core}$ [m/s]	$v_{\theta skin}$ [m/s]	v_{xbulk} [m/s]	v_{ybulk} [m/s]	v_0 [m/s]
1	-0.01	0.02	0.05	40,000	1,000	-5,500	-4,500	60,000
2	-0.02	0.02	0.03	22,000	600	-5,500	-4,500	60,000
6	0.01	-0.01	0.01	20,000	5,000	-5,500	-4,500	65,000
Average	-0.01	0.01	0.03	27,333	2,200	-5,500	-4,500	61,667
σ	0.017	0.017	0.02	11,015	2,433	0	0	2,887

*Approximate jet center at (0.038, 0.050, 0.54) [m] and approximate radius of 0.04 [m]

between the azimuthal velocity fit parameters in the toroidal and poloidal planes again show agreement within the standard deviation; however, the fits in the toroidal plane require modest bulk velocities which are not required in the poloidal plane fits. In all cases the approximate jet radius obtained from gaussian fits to the binned intensity (Figures 5.17 and 5.18) appears to undershoot the required model radius by $2-4$ cm. The resulting radius of $\sim$7.5cm is also $\sim$3 cm larger than the jet radius calculated from the high speed Kirana camera images (figure 5.23), suggesting the possibility of a diffuse outer shell that is not immediately apparent in camera images.

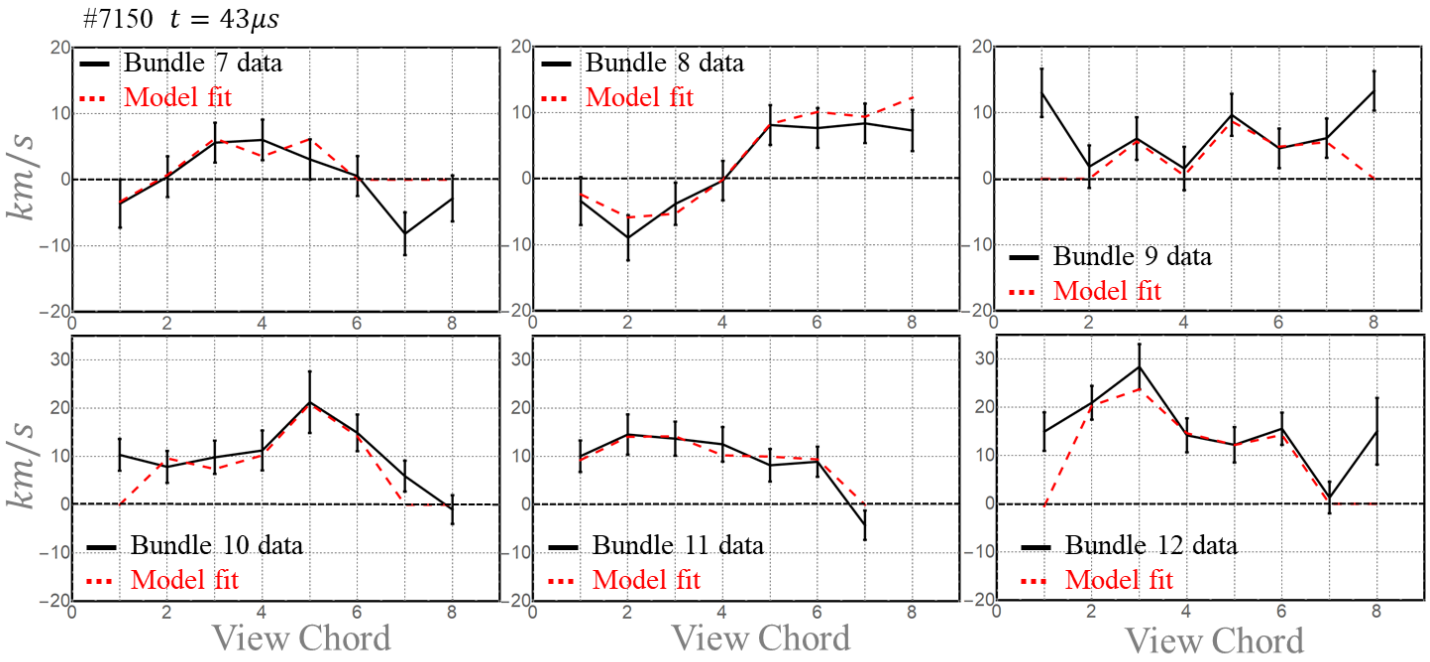


Figure 5.20. Double shell model fits to the line integrated velocities in the toroidal plane for shot #7150 at $t = 43\mu s$.

Table 5.3
Model fit parameters for shot #7150 in the toroidal plane at $t = 43\mu s$*

Bundle	Δx [m]	Δy [m]	Δa [m]	$v_{\theta core}$ [m/s]	$v_{\theta skin}$ [m/s]	v_{xbulk} [m/s]	v_{ybulk} [m/s]	v_0 [m/s]
7	0.033	-0.02	0.03	14,000	2,000	-5,500	-4,500	62,000
8	0.01	-0.022	0.04	19,000	7,000	-5,500	-4,500	62,000
9	0	-0.04	0.005	12,500	0	-5,500	-4,500	62,000
10	-0.09	-0.03	0.025	18,000	2,000	-10,000	2,000	62,000
11	-0.12	-0.08	0.08	18,200	5,800	-5,500	0	62,000
12	-0.12	-0.0835	0.08	55,000	12,000	-8,000	1,000	62,000
Average	-0.048	-0.046	0.043	22,783	4,800	-6,667	-1,750	62,000
σ	0.070	0.029	0.031	15,994	4,391	1,915	3,078	0

*Approximate jet center at (0.038, 0.050, 0.54) [m] and approximate radius of 0.04 [m]

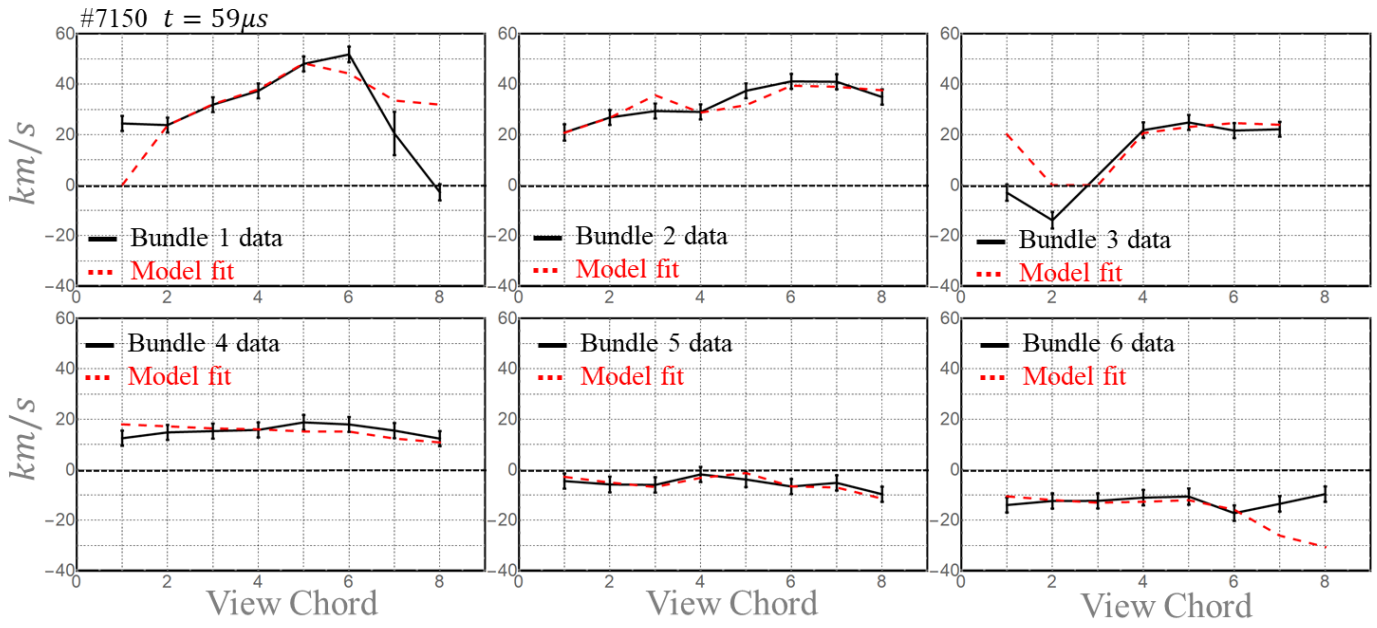


Figure 5.21. Double shell model fits to the line integrated velocities in the poloidal plane for shot #7150 at $t = 59\mu s$.

Table 5.4
Model fit parameters for shot #7150 in the poloidal plane at $t = 59\mu s$*

Bundle	Δx [m]	Δy [m]	Δa [m]	$v_{\theta core}$ [m/s]	$v_{\theta skin}$ [m/s]	v_{xbulk} [m/s]	v_{ybulk} [m/s]	v_0 [m/s]
1	-0.01	0.01	0.04	50,000	10,000	0	0	62,000
2	0	0	0.02	30,000	10,000	0	0	60,000
3	0.04	-0.04	0	20,000	17,000	0	0	60,000
4	0.035	-0.04	0.02	30,000	16,000	0	0	50,000
5	0.02	-0.03	0.01	30,000	0	0	0	53,000
6	0.03	0	0.02	33,000	3,400	0	0	41,600
Average	0.019	-0.017	0.018	32,167	9,400	0	0	54,433
σ	0.020	0.023	0.013	9806	6,729	0	0	7,818

*Approximate jet center at (-0.047, -0.024, 0.53) [m] and approximate radius of 0.035 [m]

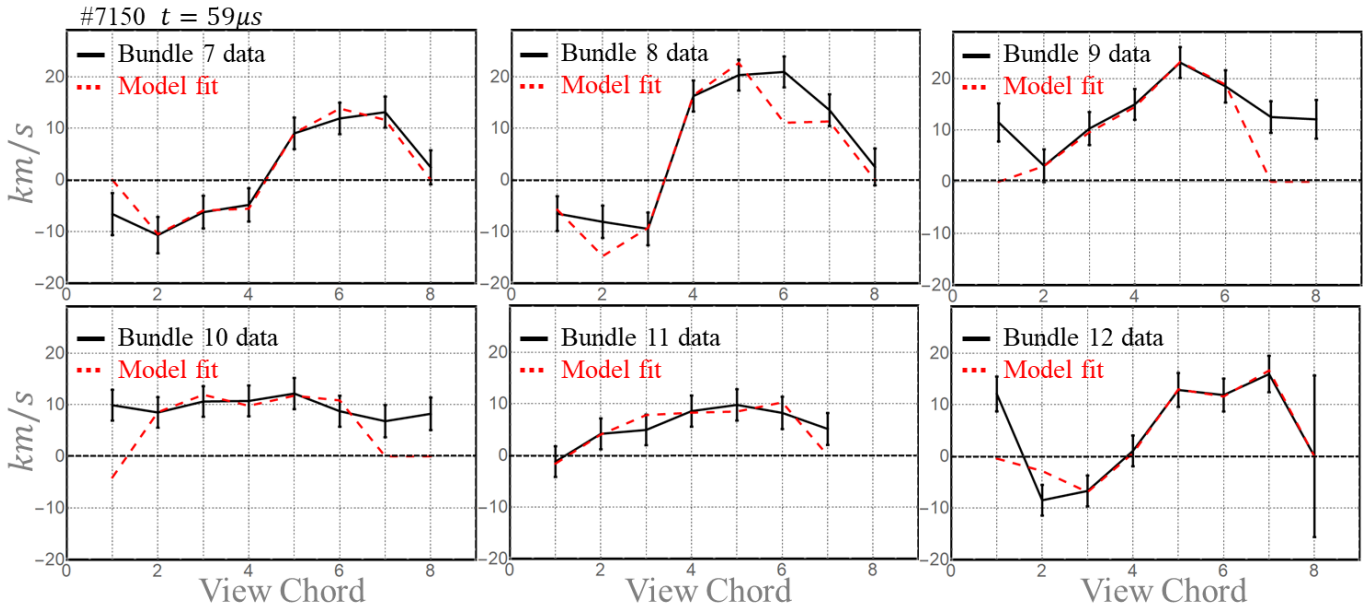


Figure 5.22. Double shell model fits to the line integrated velocities in the toroidal plane for shot #7150 at $t = 59\mu s$.

Table 5.5

Model fit parameters for shot #7150 in the toroidal plane at $t = 59\mu s$*

Bundle	Δx [m]	Δy [m]	Δa [m]	$v_{\theta core}$ [m/s]	$v_{\theta skin}$ [m/s]	v_{xbulk} [m/s]	v_{ybulk} [m/s]	v_0 [m/s]
7	0.03	-0.025	0.03	21,000	11,000	0	0	55,000
8	0.04	-0.02	0.04	50,000	10,000	-2,000	0	55,000
9	-0.018	0	0.02	13,000	8,000	-8,000	-2,000	70,000
10	0.026	-0.04	0.071	20,000	10,000	-3,000	0	55,000
11	0.02	-0.02	0.05	12,000	6,000	-4,000	2,000	55,000
12	0.04	-0.01	0.03	23,000	9,000	-6,500	6,000	55,000
Average	0.023	-0.019	0.040	23,167	9,000	-3,917	1,000	57,500
σ	0.022	0.014	0.018	13,877	1,789	2,940	2,757	6,124

*Approximate jet center at (-0.047, -0.024, 0.53) [m] and approximate radius of 0.035 [m]

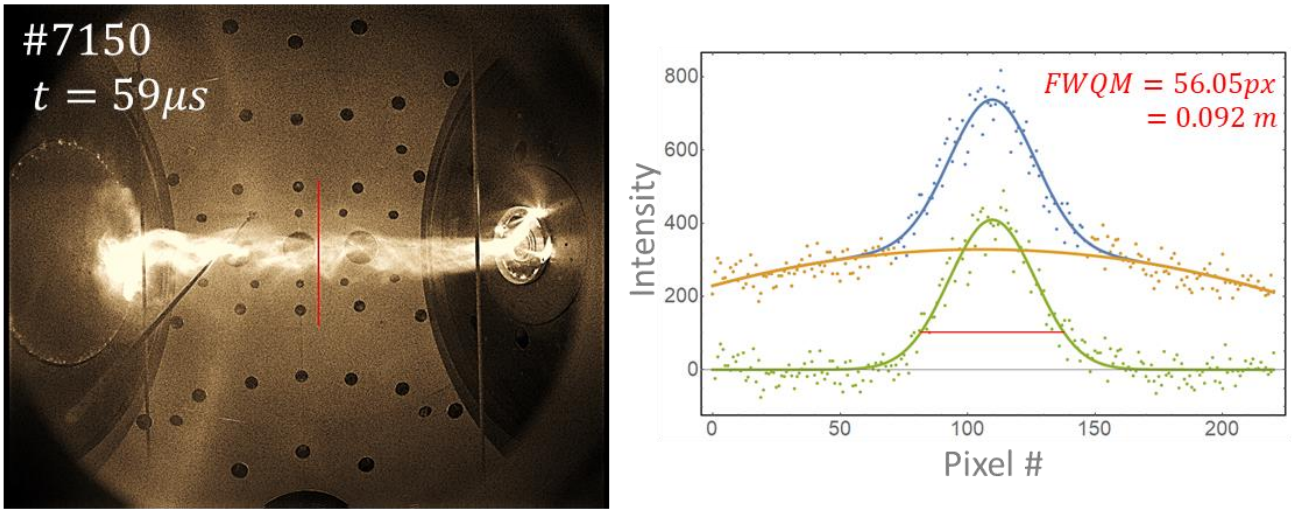


Figure 5.23. Jet diameter near the central midplane for shot #7150 at $t = 59\mu s$ calculated using the full width at quarter maximum of a gaussian fit to the intensity profile along the red line in the camera image. The resulting jet radius of 4.6 cm is larger than the initial guess for the double shell model but less than the average radius required by the model fits suggesting the possibility of a diffuse outer shell that is not apparent in camera images.

5.4.2 Abel inversion

To corroborate the results of the double shell model, another estimate of the local azimuthal velocity profile is obtained using a matrix Abel inversion technique [93, 94]. In this method the plasma volume is partitioned into radial zones with a uniform emissivity, E_j, and azimuthal velocity, v_j, with one zone for each viewing chord. Each viewing chord is assumed to be orthogonal to the jet axis such that the line integrated velocity, u_i, is only due to azimuthal rotation. Furthermore, the inversion does not account for any bulk motion of the plasma column due to oscillation. There are two free parameters Δx and Δy corresponding to the location of the jet center in the central midplane. For each viewing chord i, the minimum perpendicular distance between the chord and this center position sets the tangency radius r_i of the shell. The measured brightness along each sightline is therefore the sum of the emissivity in each zone j times the path length through that zone. A length matrix can thus be constructed which relates the emissivity vector to a brightness vector, $B_i = \int B_i^{\lambda}(\lambda) d\lambda$, by matrix multiplication

$$B_i = \sum_j L_{ij} E_j \tag{5.15}$$

such that the emissivity of each shell can be calculated by inverting the length matrix

$$E_j = \sum_i L_{ji}^{-1} B_i \, . \tag{5.16}$$

To create the length matrix the outer radius of each radial zone, s_j, is placed midway between the adjacent viewing chords such that $s_j = (r_j + r_{j+1})/2$, while the outermost boundary is given by $s_{outer} = r_{outer} + (r_{outer} - s_{outer-1})$. The resulting length matrix is diagonal with the elements computed as follows:

$$\text{For } s_j > r_i, \quad i = j, \qquad L_{ij} = 2\sqrt{s_j^2 - r_i^2} \tag{5.17}$$

$$\text{For } s_j > r_i, \quad i \neq j, \qquad L_{ij} = 2\sqrt{s_j^2 - r_i^2} - 2\sqrt{s_{j-1}^2 - r_i^2} \tag{5.18}$$

$$\text{For } s_j < r_i, \qquad L_{ij} = 0 \, . \tag{5.19}$$

Defining a new matrix $M_{ij} = L_{ij} cos\theta_{ij}$ where $\cos(\theta_{ij}) = r_i/r_j$ and θ_{ij} is angle from tangency for each emitting shell of radii r_j and the line of sight i (note that $\theta_{ii} = 0$), the product of the brightness and the line integrated velocity is

$$B_i u_i = \sum_j M_{ij} E_j v_j. \tag{5.20}$$

The emissivity weighted velocity in each shell can therefore be obtained by inverting the modified length matrix

$$E_j v_j = \sum_i M_{ji}^{-1} B_i u_i. \tag{5.21}$$

Combining equations 5.21 and 5.16, the local velocity in each shell can be calculated

$$v_j = \frac{E_j v_j}{E_j} = \frac{\sum_i M_{ji}^{-1} B_i u_i}{\sum_i L_{ji}^{-1} B_i}. \tag{5.22}$$

This analysis can be performed in both directions, for sightlines above and below the jet center. The analysis also considers errors associated with the measured line integrated doppler shifts (equation 5.9) and errors in the measured spectra

$$\sigma_{B_i^\lambda} = \beta\left(1 + \sqrt{\frac{B_i^\lambda - B_{ibackground}^\lambda}{B_{ibackground}^\lambda}}\right) \tag{5.23}$$

where β is the standard deviation of the signal in a wavelength region where there is no signal and $B_{ibackground}^\lambda$ is the mean value of this background (i.e. the uncertainty scales with the square root of the intensity, corresponding to a gaussian error distribution [95]).

Figure 5.24 demonstrates the results of this inversion technique when applied to the double shell model outlined in the previous section. In this case, the model is aligned with the $\hat{z}$ axis with a radius of 5 cm. The azimuthal velocity is 20 km/s in the core and 10 km/s in the skin and there are no axial flows. With the limited number of view chords, the reconstruction clearly has difficulty resolving the sharp transitions between the boundaries of the model. The inversion also appears to underestimate the velocities in the core and skin regions of the jet, and particularly underestimates the velocity in the core for lines of sight below the central axis of the jet. Overall however, the reconstructed profiles agree well and even reflect the flat velocity profile in the core of the jet.

Figure 5.25 shows the inverted velocity profiles for shot #7150 at $t = 59\mu s$ for bundles 7 and 8. Because the jet was well centered for these bundles and the line integrated velocities crossed through zero only once, the inversion was carried out from both sides while the two free parameters Δx and Δy were adjusted from the initial guess obtained from the double shell model (Figure 5.18) until the magnitudes of two innermost velocities agreed. For bundle 7, this velocity was $\sim$35km/s while for bundle 8 it was between 55 and 80 km/s, both corresponding to a $\Delta x = 0.04$ m and a $\Delta y = -0.03$ m. In both cases, the inverted local velocities suggest azimuthal rotation in the counter-clockwise direction with blue shifts for lines of sight above the jet center and red shifts for lines of sight below. The magnitude of the velocities is larger than those estimated using the simple double shell model; however, the ratio of the core velocities agrees. Furthermore, for both cases significant velocity shear is observed. For bundle 8, the velocity monotonically decreases in magnitude to zero at the outer view chords. For bundle 7, the trend is less obvious as there appears to be a dip in velocity on both sides of the jet center, increasing again to $\sim$35 km/s before dropping to zero at the outer view chords.

For the remaining bundles which were not as well centered or had line integrated velocities crossing through zero more than once, the inversion was carried out for view chords below the central plasma column only while Δx and Δy for each bundle was adjusted until the

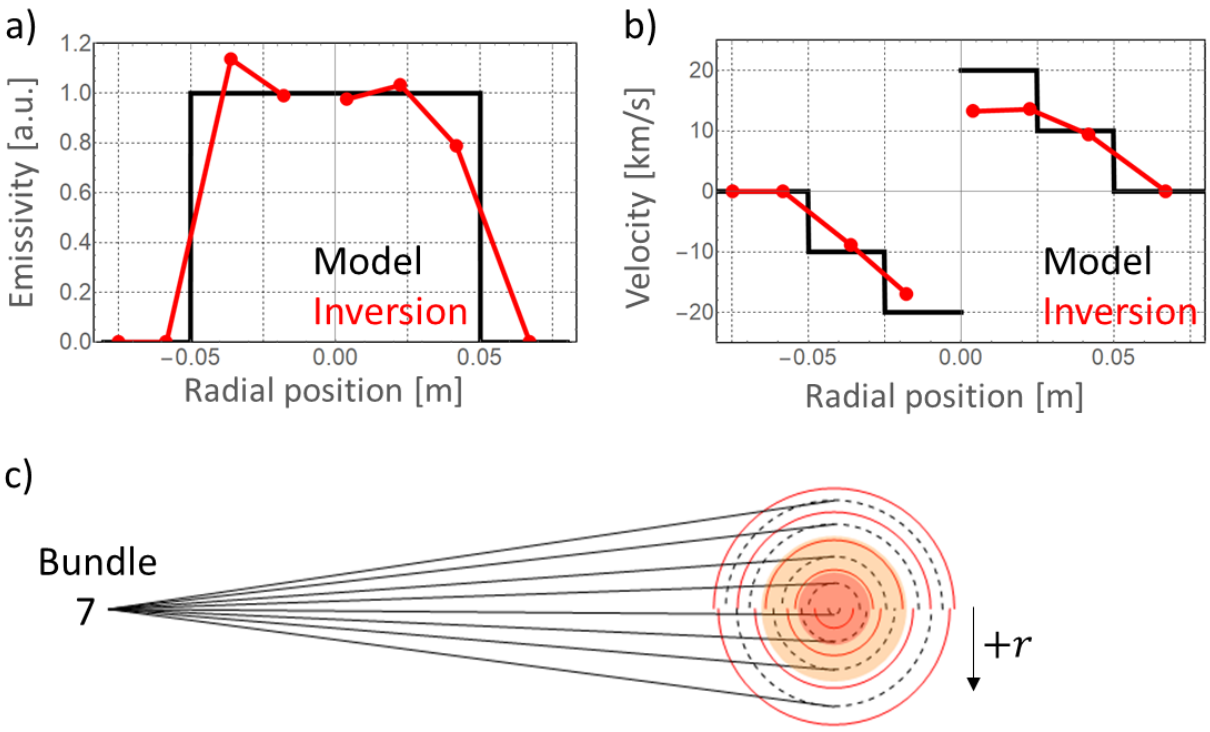


Figure 5.24. Test of the matrix Abel inversion method using the double shell model with 20 km/s azimuthal velocity in the core, 10 km/s azimuthal velocity in the skin, and no axial flows. The model is presumed to be perfectly aligned with the $\hat{z}$ axis and bundle 7 is used for the inversion. a) Comparison of the inverted emissivity profile to the uniform emissivity profile of the double shell model. b) Comparison of the inverted azimuthal velocity profile to the double shell model. c) Graphical representation of the shells used in the inversion technique compared to the placement of the double shell model with a skin shell shown in orange and the core shell shown in red. Dashed lines represent the tangency radii of a given chord while red lines delineate the separate shells.

profiles were good agreement (Figure 5.26). Apart from fiber bundle 12, the inverted local velocities all decrease monotonically from the central axis where the velocity is on the order of $50 - 75$ km/s and approach zero for $r \gtrsim 0.07$m. The strong red shifts again suggest azimuthal rotation in the counter clockwise direction with a peak azimuthal velocity in the core larger than predicted with the double shell model. Although the error bars are significant in all cases, and the inversion does not account for any bulk velocity of the central plasma column, the results appear to corroborate the presence of strongly sheared azimuthal flows.

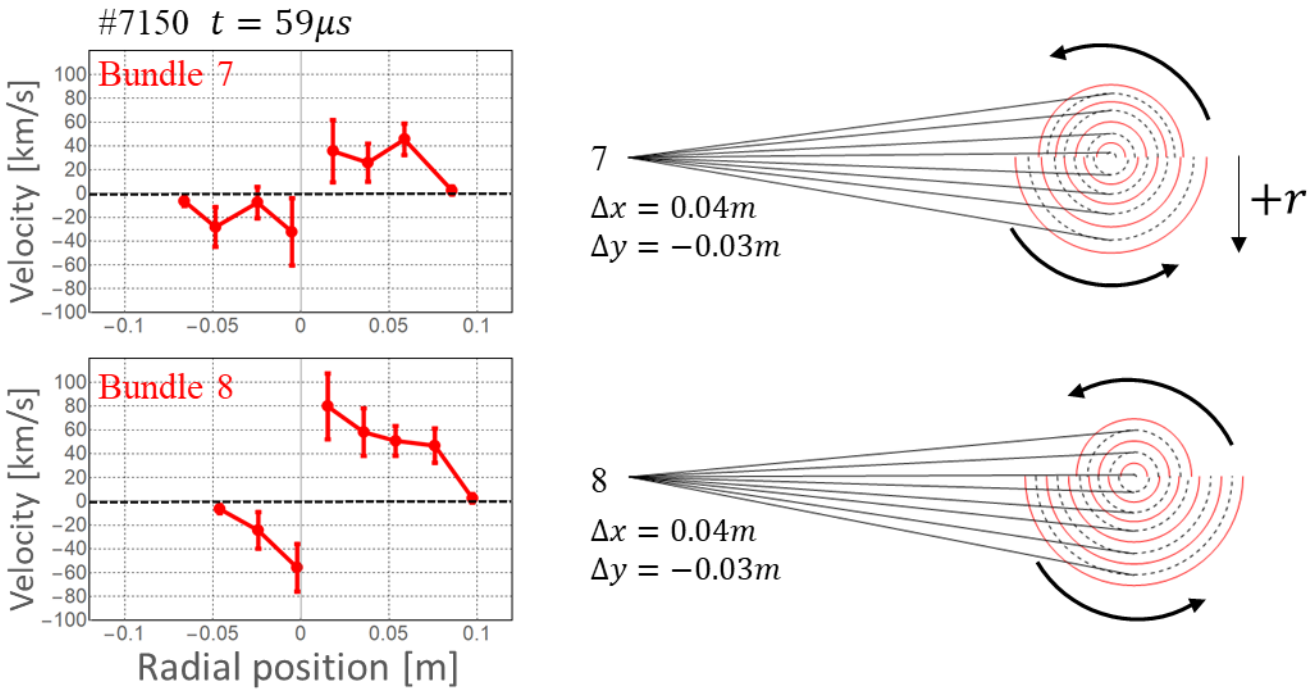


Figure 5.25. Local azimuthal velocities for shot #7150 at $t = 59\mu s$ for bundles 7 and 8 using the matrix Abel inversion technique. The inversion was carried out for lines of sight above and below the center of the jet. The two free parameters, Δx and Δy, were adjusted from the initial guess location $(x, y, z) = (-0.047, -0.024, 0.53)$ [m] obtained using gaussian fits to the binned intensity (Figure 5.18) until the magnitude of the velocity for the inner two view chords agreed.

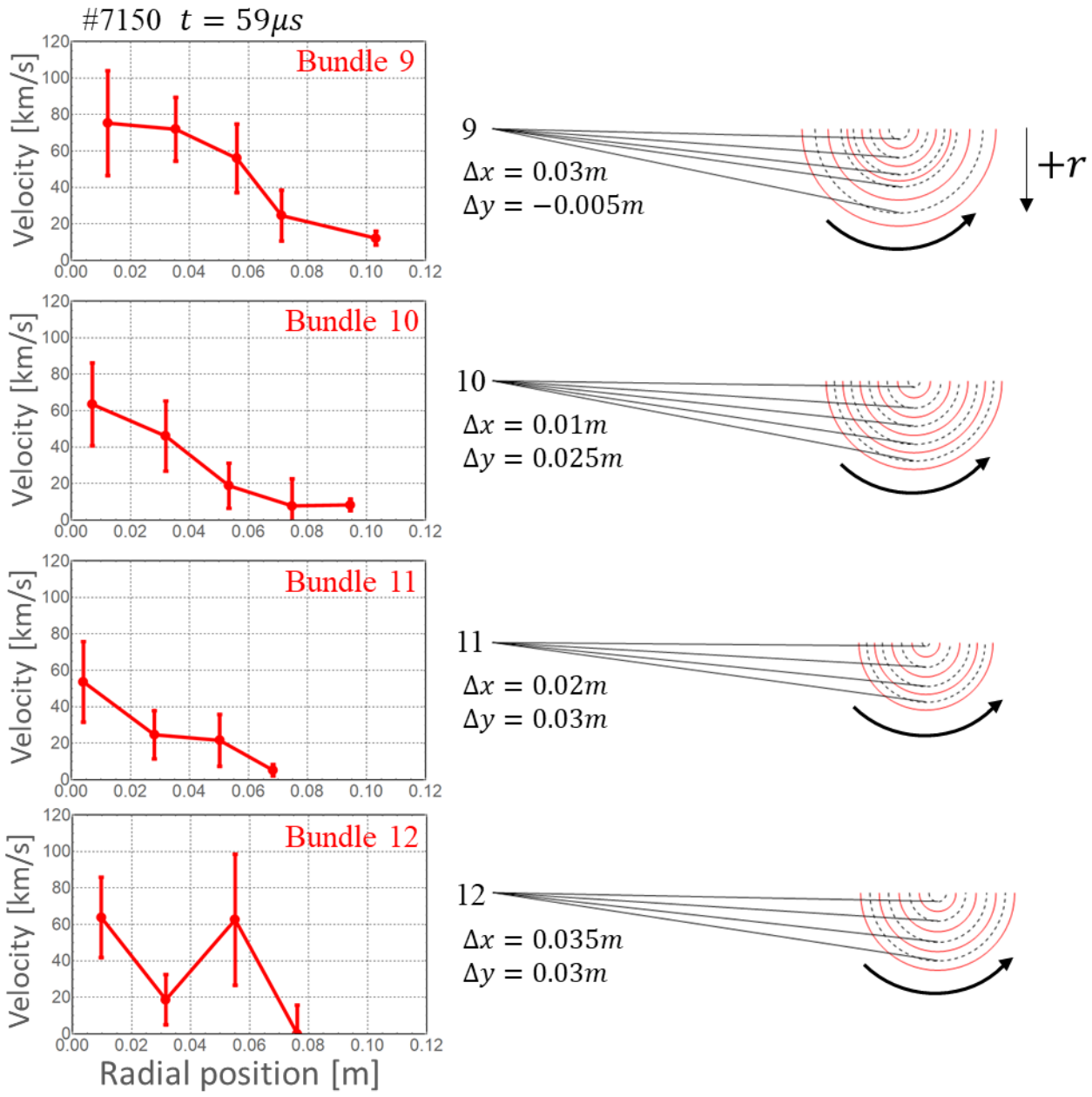


Figure 5.26. Local azimuthal velocities for shot #7150 at $t = 59\mu s$ for bundles 9-12 using the matrix Abel inversion technique. The inversion was carried out for lines of sight below the center of the jet only. The two free parameters, Δx and Δy, were adjusted from the initial guess location $(x, y, z) = (-0.047, -0.024, 0.53)$ [m] obtained using gaussian fits to the binned intensity (Figure 5.18) until the profiles were in good agreement.

5.5 Discussion

This chapter presents the results obtained from IDS measurements on the MoCHI.LabJet experiment using a custom, multichannel optical fiber bundle. Observations using the H_α emission line show line integrated axial flows for neutral hydrogen on the order of ~ 10 km/s with possible radial outflows on the same order of magnitude, manifested as a systematic blue shift across the entire data set. In the toroidal plane, several bundles suggest the presence of sheared azimuthal flows with differences in line integrated velocities between chords on the order of ~ 10 km/s; however, the profiles are not consistent with what would be expected for coherent azimuthal rotation. Temperatures calculated from spectral line broadening indicate significant heating of the long, stable jet from ~ 5eV to ~ 20eV despite an overall decrease in jet current during this time of ~ 90kA. The presence of a hollow temperature profile observed in the toroidal plane suggests that the skin of the plasma column is hotter than the core. These observations agree with estimates obtained using the gun current and voltage waveforms assuming Spitzer resistivity (Figure 4.17). Based on estimates of the magnetic field strength of $\sim$0.5 T, temperatures of $20eV$ correspond to a plasma $\beta = 2\mu_0 n k_b T/B^2 \approx 1$.

Similar results were obtained for Nitrogen plasmas, where line integrated axial velocities were also on the order of ~ 10 km/s but with no systematic skewing of the data set due (presumably) to radial outflows. Variation in the line integrated toroidal velocities on the order of $\sim 5-10$km/s also suggest the presence of azimuthal shear flows but again were not reflective of a coherent rotation. Observations using CII impurity lines exhibited qualitatively different dynamics, with significantly enhanced line integrated axial velocities on the order of $40-50$ km/s and line integrated azimuthal velocities upwards of 20 km/s with clear indications of coherent azimuthal rotation. We believe these observations best reflect the dynamics of the hydrogen ions within the jet and therefore utilized several techniques to deconvolve the local velocities from this data. Based on the first technique, which fit line integrated velocities through a simple double shell model to the line integrated experimental data, local axial velocities on the order of ~ 60 km/s are predicted, with slight deceleration between $t = 43\mu s$ and $t = 59\mu s$ as the jet piles up against the end-wall and the driving current decreases. Fits in the toroidal measurement plane suggest azimuthal velocities in the core of the jet on the order of $\sim$20 km/s with velocities in the skin increasing from $\sim$5 km/s to $\sim$10 km/s between frames. The direction of the inferred rotation is in the counter-clockwise ($+\hat{\phi}$) direction, which is consistent with the predicted $\vec{E} \times \vec{B}$ direction

of the gun where the radial electric field points inward and the magnetic field points out of the gun. Based on the model fit parameters, the jet radius is approximately 7.5 cm, which is slightly larger than measurements based on the full-width-quarter max of gaussian fits to the intensity profile of camera images in the visible wavelength (Figure 5.23). This further suggests the possibility of a hotter, flowing outer shell for which visible light emission is reduced.

The flow shear estimates obtained using the double shell model can be compared to the theoretical threshold for wall-free stabilization [36]

$$v' > 0.1kv_a \tag{5.24}$$

where v' is the flow shear, k is the mode number, and v_a is the Alfvén velocity. Although initially developed for axial shear flow stabilized Z-pinch plasmas where $v' = dv_z/dr$ and $k = n\pi/L$, the same phase mixing physics responsible for stabilization is assumed to apply in the azimuthal direction, such that $v' = dv_\phi/dr$ and $k = m\pi/2\pi a$ is the azimuthal mode number. With a line integrated density of $\sim 7 \times 10^{20} [m^{-2}]$ (figure 4.16c) a radius of ~ 7.5 cm and a magnetic field strength of $\sim$0.4T (see the following chapter), the Alfvén velocity is $\sim$90 km/s and the required azimuthal flow shear threshold (for $m = 1$) is $\sim$60,000 $[s^{-1}]$. Based on the average azimuthal velocity in the core of 23 km/s and in the skin of 9 km/s at $t = 59\mu s$, the resulting flow shear $v' \approx \Delta v/(\mathrm{a}/2) \approx 380{,}000\ [s^{-1}]$ is $\sim$6 times larger than the stabilization threshold in the azimuthal direction. Furthermore, the peaked velocity profile agrees with results from a linear z-pinch stability analysis which found that azimuthal velocity profiles in the approximate functional form of a vortex $(v_\phi(r) = v_0/(a^2 + r^2)^{\frac{1}{2}})$ reduced the kink instability growth rate most efficiently [96]. In the axial direction, the shear flow threshold for the 1.1m long jet (with $n = 1$) is $\sim$26,000 $[s^{-1}]$. Taking the derivative of the model axial flow profile (equation 5.13) using the average fit values in the poloidal plane at $t = 59\mu s$ (Table 5.4) this velocity shear is exceeded for $r \gtrsim 1$ cm. In fact, a maximum axial velocity shear of $\sim 4 \times 10^6$ $[s^{-1}]$ at $0.88a$ could, in theory, stabilize axial modes up to $n \approx 150$.

To corroborate these azimuthal flow shear estimates, a matrix Abel inversion was performed on the same data set. The results of this inversion also suggested strong azimuthal shear flows but with significantly enhanced velocities compared to the double shell model. In fact, local azimuthal velocities in the core of the jet of 50-80 km/s are of the same order of magnitude as the axial flows predicted using the double shell model. A Linear fit to the collective inverted velocity

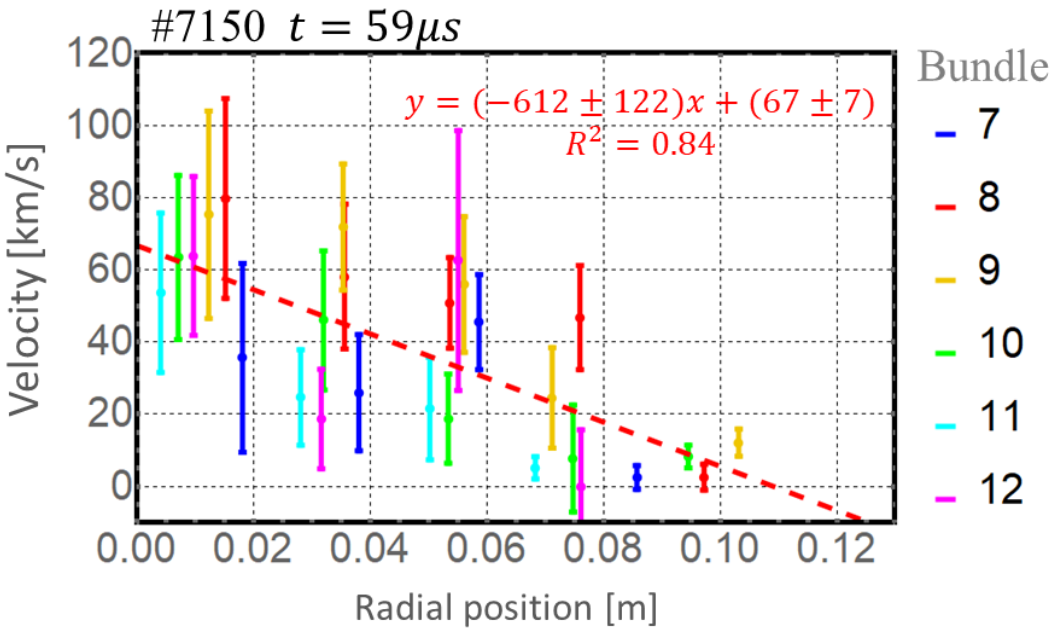


Figure 5.27. Linear fit to the collective toroidal velocity profiles obtained using a matrix Abel inversion technique for shot #7150 at $t = 59\mu s$. The results indicate a peaked azimuthal velocity profile with a shear of $\sim -6 \times 10^5 s^{-1}$ and azimuthal velocities on the order of ~ 65 km/s in the core of the jet.

profiles for bundles 7-12 (Figure 5.27) suggests an azimuthal flow shear of ~600,000 $[s^{-1}]$, ten times the required stabilization threshold value.

These results suggest that the long stable jets observed in the MoCHI.Labjet experiment exhibit enhanced stability due to the presence of strong helical shear flows. For shot #7150 at $t = 59\mu s$, the jet is collimated, straight, and stable despite occupying classically unstable space (Figure 4.16d) with $\lambda_{gun}/4\pi L \approx 2.8$. An instability only appears at the end of the discharge when the gun current is approaching zero (Figure 4.16a frame 9). However, a classical current carrying magnetic flux tube with no flows should be more stable with decreasing current. Since the current is ultimately responsible for the $\vec{j} \times \vec{B}$ forces that drive stabilizing helical flows, in competition with destabilizing $\vec{j} \times \vec{B}$ magnetic forces, this suggests that the stabilizing effect of helical shear flows may weaken more rapidly than the destabilizing effect of helical magnetic fields as the current decreases. This would explain why, in general, the long stage III jets go unstable near the end of the discharge despite their return towards classical stability space.

Chapter 6
MAGNETIC FIELD MEASUREMENTS ON THE MOCHI.LABJET EXPERIMENT

In addition to flow velocity measurements, knowledge of the magnetic field structure within a plasma is critical towards understanding its evolution, dynamics, and stability. Prior to the implementation of three custom magnetic probe arrays, magnetic field strengths in the MoCHI experiment were estimated using two methods. In the first method, electron densities were measured using an unequal pathlength heterodyne interferometer while the axial propagation velocity of the initial jet was assumed to be of the order of the Alfvén velocity. Based on an electron density of 2.5×10^{22} $[m^{-3}]$ and an axial propagation velocity of 70.6 km/s for the stage I jets, the resulting magnetic field $B = v_a\sqrt{\mu_0 m_i n_i}$ was calculated to be ~0.51 T. Assuming all the gun current flows axially through the jet, the second method is to add the contribution of the toroidal magnetic field to the poloidal bias field. For gun currents of ~60 kA and jet diameters of ~5 cm, the resulting toroidal field is 0.48 T. With a bias magnetic field of ~0.3 T at the gun surface, the resulting total magnetic field of ~0.56 T agrees well with the results from the previous method. Neither of these methods however, provide any information regarding the structure and dynamics of the magnetic fields during the evolution of the jet. For the purposes of stability analysis, the jet was therefore assumed to resemble a simple azimuthally symmetric screw pinch with a stability described by the Kruskal-Shafronov condition (equation 4.1 and equation 4.4). To better resolve the strength, structure, and dynamics of magnetic fields in the MoCHI.LabJet experiment, a set custom magnetic probe arrays were constructed and calibrated in house. This chapter presents the results from these diagnostics and suggests that the long stable jets have a more complicated structure than a simple azimuthally symmetric screw pinch.

6.1 Theory and calibration

According to Faraday's law, any time changing magnetic flux (ψ) through a coil will induce a voltage given by

$$V_{coil} = \oint \vec{E} \cdot d\vec{l} = -\iint \frac{\partial \vec{B}}{\partial t} \cdot d\vec{S} = -\frac{\partial \psi}{\partial t}. \tag{6.1}$$

Specifically, for a coil with N turns and a cross sectional area A, the average magnetic field threading the loop can be calculated by integrating the coil voltage with respect to time

$$B = \frac{1}{NA}\int_0^t V_{coil}\, dt \ . \tag{6.2}$$

For each magnetic probe array (Figure 4.6), 135 commercially available inductor chips (Coilcraft 1008CS-472XGLB) were grouped into 45 orthogonal clusters ($\hat{x}$, $\hat{y}$, $\hat{z}$), 25 in a high-resolution section at the center of the array (1.1 cm spacing between clusters) bordered by two low resolution sections of ten clusters (3.3 cm spacing). Each section and cluster is numbered, with an increasing value as one moves from the end of the array inward (i.e. chip 1.1 on the far side to chip 3.10 on the near side). Because the exact *NA* of each chip is not known, calibration to a known magnetic field is required to obtain accurate measurements. For our known field, we use a custom 3D printed 3-axis Helmholtz coil. Each axis of the coil has a known number of turns (n) and known radius (a) through which we can apply a known current (I). The field at the center of the Helmholtz coil is given by

$$B_{applied} = \left(\frac{4}{5}\right)^{\frac{3}{2}}\frac{\mu_0 nI}{a} \tag{6.3}$$

such that the time-changing field is

$$\frac{\partial B_{applied}}{\partial t} = \left(\frac{4}{5}\right)^{\frac{3}{2}}\frac{\mu_0 n}{a}\frac{\partial I}{\partial t}. \tag{6.4}$$

Equating the coil voltage to the resulting change in flux and solving for NA one obtains

$$NA = -\frac{a}{\left(\frac{4}{5}\right)^{\frac{3}{2}}\mu_0 n}\frac{\int V_{coil}dt}{\int dI}. \tag{6.5}$$

Provided that all the coils are aligned perfectly along the $\hat{x}$, $\hat{y}$, and $\hat{z}$ axis, equation 6.5 would be sufficient to obtain a calibration NA value for each inductor chip. However, since the chips are mounted on a Delrin stalk and are not perfectly aligned, it is necessary to construct a

calibration matrix $\mathbb{A}$ for each cluster such that $B_{applied} = \mathbb{A}B_{measured}$. To construct this calibration matrix, we follow the procedure outlined in appendix C of reference 80. This approach uses a separate calibration matrix $\mathbb{M}$ which accounts for orientation errors and includes NA explicitly such that $B_{applied} = \mathbb{M}\vec{A}$, where

$$\vec{A} = \begin{pmatrix} \int V_x dt \\ \int v_y dt \\ \int v_z dt \end{pmatrix} \tag{6.6}$$

and $\mathbb{M}$ can be calculated by noting that $\mathbb{M}^{-1}B_{applied} = \vec{A}$. Solving for $\mathbb{M}^{-1}$ one obtains

$$\mathbb{M}^{-1} = \begin{pmatrix} \int V_x dt/B_{xapplied} & \int V_x dt/B_{yapplied} & \int V_x dt/B_{zapplied} \\ \int V_y dt/B_{xapplied} & \int V_y dt/B_{yapplied} & \int V_y dt/B_{zapplied} \\ \int V_z dt/B_{xapplied} & \int V_z dt/B_{yapplied} & \int V_z dt/B_{zapplied} \end{pmatrix} \tag{6.7}$$

where the applied magnetic fields are the known fields due to the Helmholtz coil. To obtain this calibration matrix, it is therefore necessary to perform a minimum of three measurement shots for each cluster for a total of 3×45 measurements for each magnetic probe array.

To account for the frequency response of the chips; however, it is beneficial to repeat these measurements at multiple frequencies of applied field. Sweeping across 11 different frequencies for all three magnetic probe arrays therefore requires 4,455 measurement shots. To accomplish such a task in a reasonable amount of time, the three axis Helmholtz coil was mounted to a sled with a stepper motor controlled by a national instruments myDAQ integrated into the MoCHI.Control using a custom LabVIEW script. Current through the Helmholtz coil was provided via a function generator with frequency control provided through the myDAQ. The current through the Helmholtz coil was measured using a Pearson current monitor connected to the FPGA digitizers while the magnetic probe signals were also measured using the FPGA digitizers at a sampling rate of $50MHz$. For each chip, 11 shots were recorded at different frequencies from 40kHz to 3.2MHz before stepping to the next chip and repeating the process. After sweeping the entire array, the direction of the applied magnetic field was changed and the process was repeated. Figure 6.1 shows a typical cluster response to the magnetic field applied in the $\hat{x}$ calibration direction. Significant pickup on the $\hat{y}$ oriented chip indicates misalignment with the calibration frame which is accounted for in $\mathbb{M}$ calibration matrix. Figure 6.2 shows the

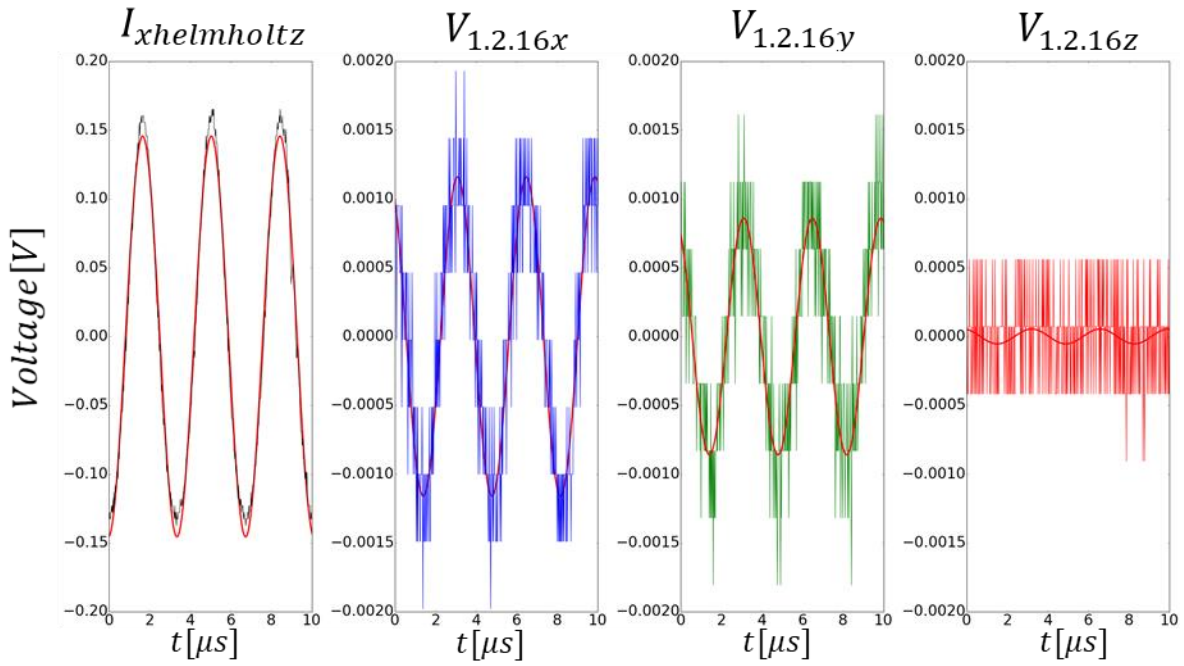


Figure 6.1. Response of cluster 2.16 of magnetic probe array 1 to a magnetic field applied in the $\hat{x}$ calibration direction at 300 kHz. Comparable pickup in the x and y chips indicates that they are not well aligned with the calibration axis. Calibration matrices constructed using this analysis correct for this misalignment.

frequency response of the main diagonals for the inverse calibration matrix and accounts not just for the inductor chip frequency response but also contributions due to the electromagnetic shielding of the stainless-steel tubing in which the Delrin stalk is encased. For chips aligned with the tubing (calibration $\hat{z}$ direction) a steeper drop-off is observed compared to the chips oriented perpendicular to the tubing. This is to be expected since the soak time in the parallel direction $\tau = \mu_0 R d/\eta$ corresponds to a lower cutoff frequency than for waves in the perpendicular direction, where the cutoff frequency is $\omega = 1/d^2 \sqrt{2\eta/\mu_0}$ [97], where R is the radius of the tubing, d is its thickness, and $\eta = 3.5 \times 10^{-6}\ \Omega \cdot m$ is the resistivity of stainless steel. For the analysis of shot data, calibration matrices obtained using the 300 kHz signal were used corresponding to a characteristic timescale of 3.3 μs. However, to examine phenomena at different time scales, the calibration matrices can easily be swapped to account for the frequency response of the probes.

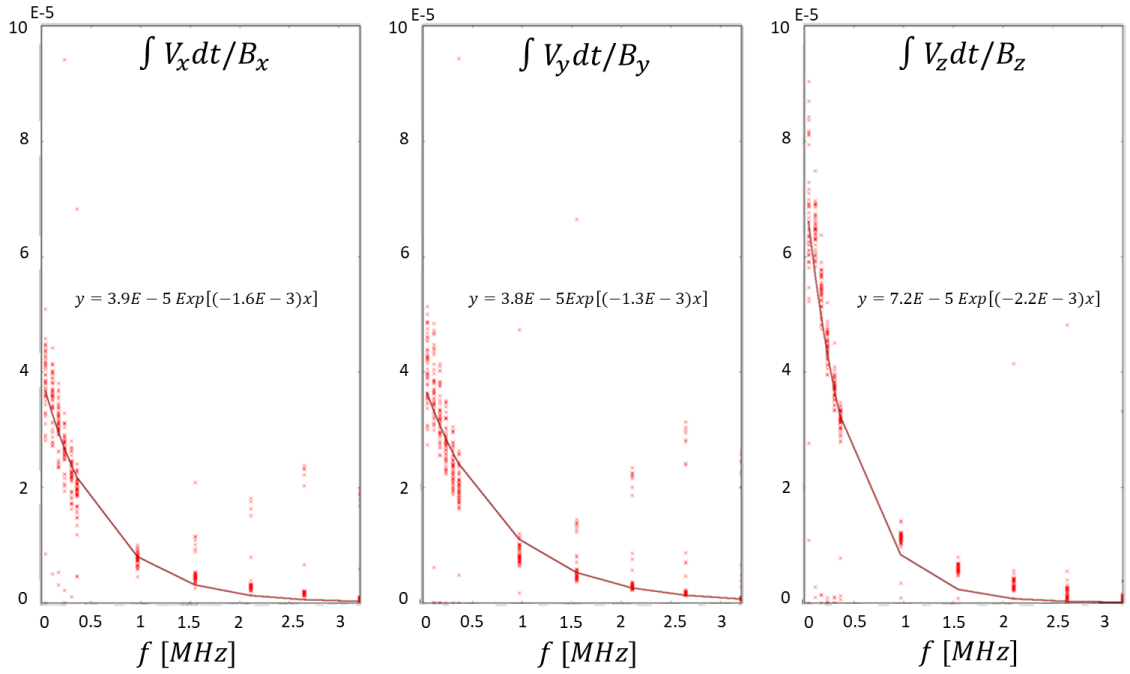


Figure 6.2. Frequency response of the main diagonals in the inverse calibration matrix for magnetic probe array 1. Calibration matrixes obtained using a 300kHz signal, corresponding to a period of $\sim 3\mu s$ were selected to be used for the analysis of shot data but can be swapped out easily to examine phenomena occurring at different timescales. Resonances at high frequencies were observed for several chips but was not systematic.

Figure 6.3 shows a reconstruction of magnetic fields applied in each orthogonal direction after application of the calibration matrix for chip 2.16 of magnetic probe array 1. The reconstruction accounts for any errors associated with uncertainty in the Helmholtz coil radii and fits to the measured Helmholtz coil current and chip voltages (i.e. calibration matrix errors). Despite the relatively weak calibration field, it is observed that the calibration matrix adequately corrects for the misalignment of the chips and provides a good reconstruction of the applied field. A small phase shift between the applied and reconstructed field is observed due to the finite transmission line distance; however, to ensure that the signs of the calibration matrix are correct, the orientation of each chip was confirmed using a single pulse through the Helmholtz coil.

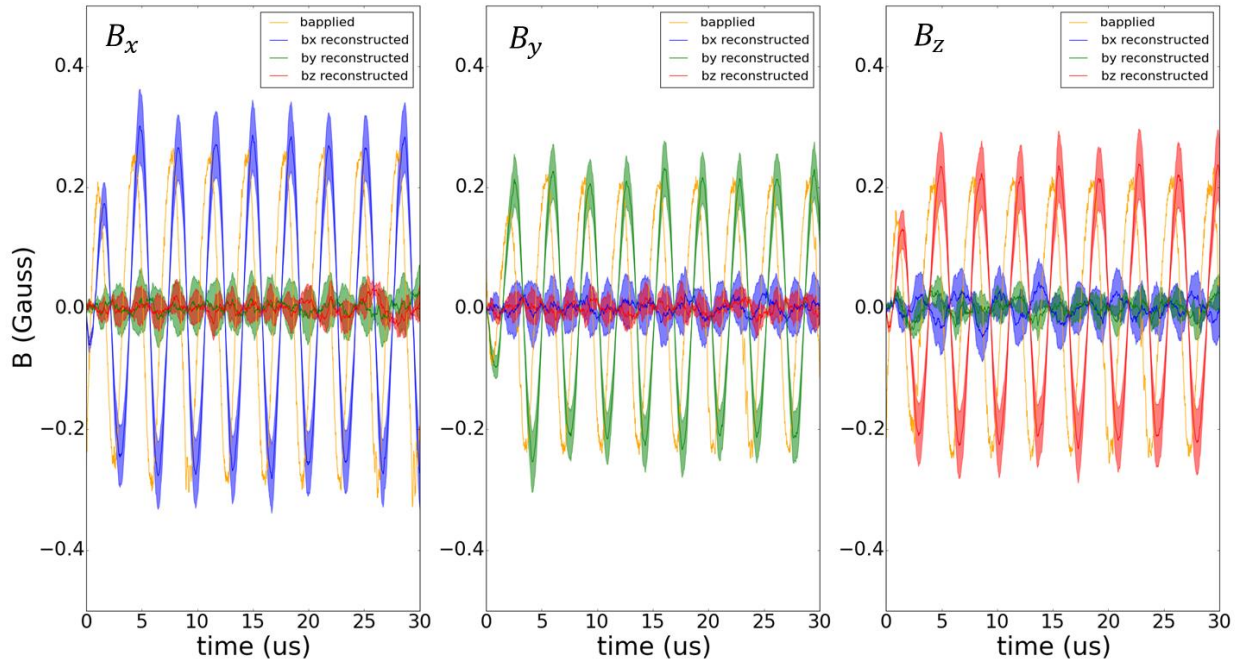


Figure 6.3. Reconstructed magnetic fields for cluster 2.16 of magnetic probe array 1 using calibration matrix for 300kHz. Despite the weak calibration field, the reconstruction agrees with the applied field and corrects for the misaligned chips

6.2. Shot Data

Magnetic probe array 1 was the first to be calibrated and was initially mounted 21 cm in front of the LabJet gun electrodes such that the center of the array (chip 2.13) was aligned with the center of the gun. Chip voltages were carried from the magnetic probe array box to the FPGA digitizers via double-shielded, twisted-pair CAT-7 cables. The custom attenuator boards were set to -40dB to prevent overvoltage of the FPGAs. 96 digitizer channels sampling at 50 MHz limited measurements to the full high-resolution center section (section 2) bordered by 4 additional clusters on the near side (section 3) and 3 clusters on the far side (section 1). Prior to integration, voltage signals are filtered with a digital third order Butterworth bandpass filter with a low frequency cutoff of 13 kHz and a high frequency cutoff of 23 MHz. The experimental coordinate frame is defined with the $\hat{x}$ direction pointing towards the magnetic probe box (mounted on the far side of the chamber in camera images), the $\hat{z}$ direction pointing out of the gun, and the $\hat{y}$ direction pointing downward to complete the right hand orthogonal coordinate system. The experiment coordinate frame is related to the calibration frame as follows:

$$\begin{aligned} \hat{x}_{exp} &= \hat{z}_{cal} \\ \hat{y}_{exp} &= \hat{y}_{cal} \\ \hat{z}_{exp} &= \hat{x}_{cal} \end{aligned} \tag{6.6}$$

such that transformation from the measured magnetic fields in the calibration frame to the experiment frame can be accomplished simply by exchanging the $\hat{x}$ and $\hat{z}$ components.

Figure 6.4 shows typical magnetic field traces obtained with this setup for shot #6999. The field strength is largest near the center of the array (chip 2.13) and decreases with increasing or decreasing chip number. The field in each orthogonal direction appears to oscillate throughout the shot and oscillates faster during stage I and stage II before entering a relatively quiescent period during the transition between stage II and stage III jets. A peak magnetic field strength of ~ 0.45 T is observed for chip 2.13 during the stage II jet and agrees well with previous estimates. However, the magnetic field during the stage I jet is less than the predicted based on the axial propagation velocity, electron density, and jet current. One potential explanation for this reduced field is explored below.

A more intuitive visualization of the magnetic field evolution can be realized by generating contour plots of the field strength in each orthogonal direction as a function of time and displacement from the gun center. Figure 6.4 and 6.5 show the results of this analysis for two representative shots during this operation, #6999 and #7016 respectively. As expected for well collimated jets, the strength of the magnetic field is larger near the central axis ($x = 0$) over the entire duration of the shot. Furthermore, this region of enhanced field is narrower for shot #7016, which appears in camera images to be slightly more collimated than shot #6999. The most intriguing and unexpected feature; however, is the reversal and simultaneous presence of strong positive and negative axial fluxes. Such a feature is not expected for a simple cylindrical screw pinch where, based on the orientation of the bias magnetic field (out of the gun), strong positive axial flux would be expected in the core of the jet, bordered by a weaker, diffuse negative return flux. These results therefore suggest that the magnetic field structure, at least near the gun electrodes, is distinctly different than a screw pinch over the formation and propagation of the stage II and stage III jets. Earlier in time however, as the stage I jet propagates past the magnetic probe array, line profiles of the magnetic field reflect a profile more indicative of a simple azimuthally symmetric screw pinch (Figure 6.7a) with a peaked axial field in the core, a weak return flux at the edge, and a reversal in B_y (equivalent to B_ϕ) at $x = 0$. For an unflared screw

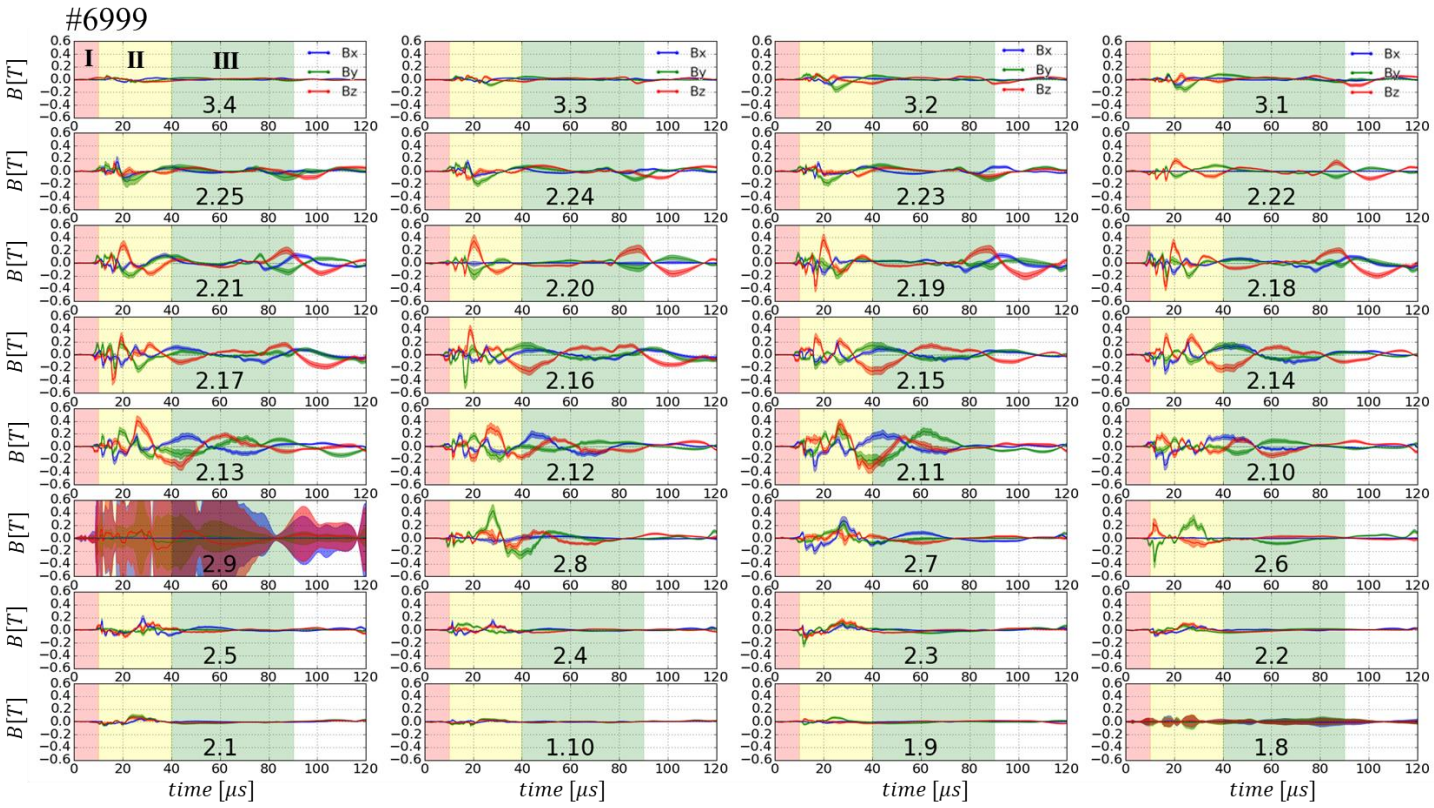


Figure 6.4. Magnetic field traces obtained from magnetic probe array 1 placed 21 cm in front of the LabJet gun for shot #6999. Filled regions indicate the associated calibration error. A bad chip for cluster 2.9 results in large errors and is therefore excluded from future analysis.

pinch B_x (equivalent to B_r) should be zero; however, because our stage I jets are finite in length, a finite B_x which reverses direction at $x = 0$ is anticipated due to the arched magnetic field at the leading edge of the jet and any flaring of the column itself. Interestingly however, the direction of the measured magnetic field for the stage I jet is exactly opposite what is predicted based on the bias field direction with an antiparallel current density (i.e. $-\lambda$). This can be demonstrated by comparing the profile of a negative helicity Lundquist solution with a positive axial field (equation 1.25, Figure 6.7c) to the observed magnetic field profile for the stage I jet (Figure 6.7a). Furthermore, the strength of the measured magnetic field is weaker than the initial estimates obtained from the jet current, axial propagation velocity, and electron densities. One potential culprit for these discrepancies is a large diamagnetic current

$$\vec{J}_D = \frac{\vec{B} \times \nabla p}{B^2} \tag{6.7}$$

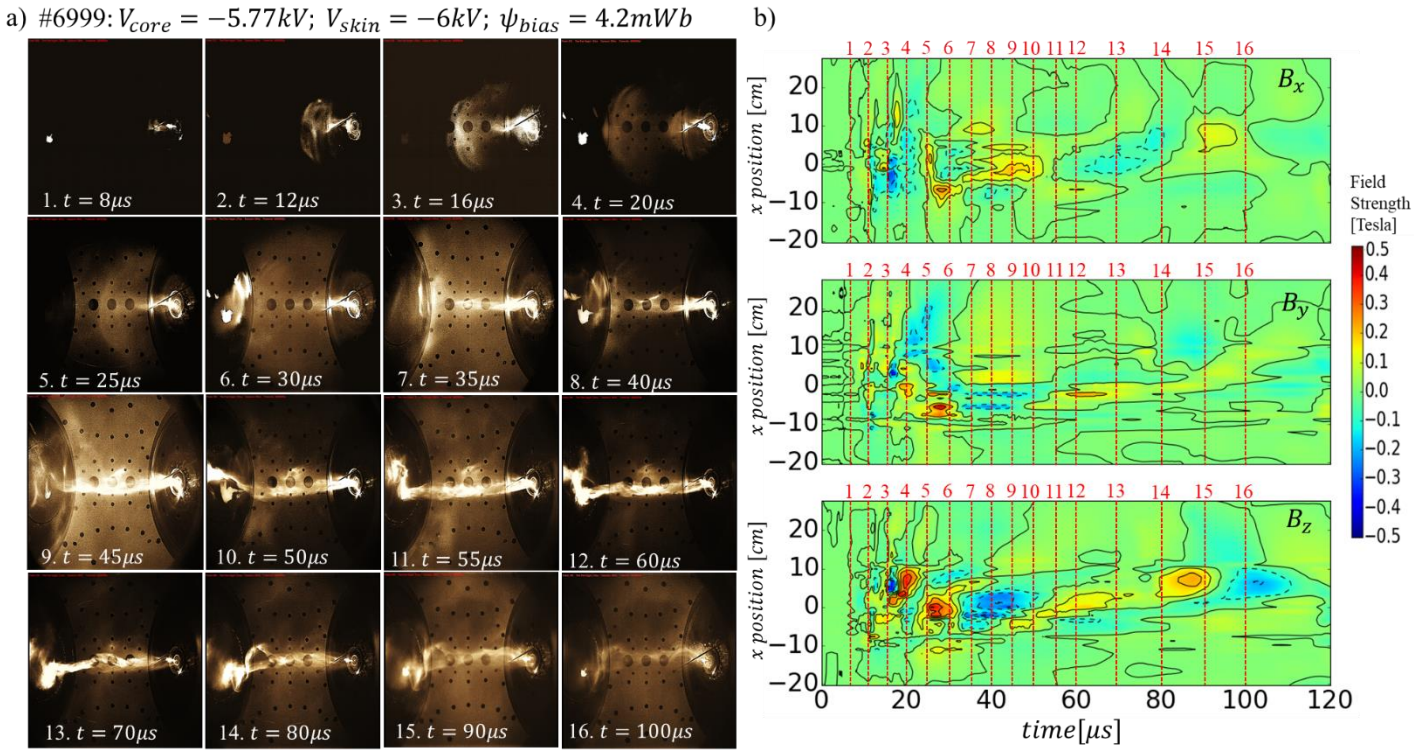


Figure 6.5. Magnetic field evolution for shot #6999 using a single linear array placed 21cm in front of the gun electrodes shows a regular reversal in the magnetic field direction in all three orthogonal directions. a) High speed images taken with the Kirana camera at 1 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system. Alternating positive and negative axial magnetic fields (B_z) with comparable magnitudes ($\sim$0.4 T) suggest that the magnetic field structure of the long stable jet is not a simple screw pinch.

where ∇p is the pressure gradient, which acts to reduce the imposed field. Because the bias field is constant over the timescale of the plasma, and is thus not measured with the $\dot{B}$ probes which measure the time changing field, if the diamagnetic field increases faster than the increasing field due to axial flux compression and current ramp up this could result in the weak, reversed magnetic field measured in the stage I jets. Regardless of the direction of the fields however, the fact that the profile resembles a screw pinch provides one possible explanation for why the stage I jets obey the Kruskal-Shafranov instability condition (Equations 4.1 and 4.4) while the stage II and stage III jets do not.

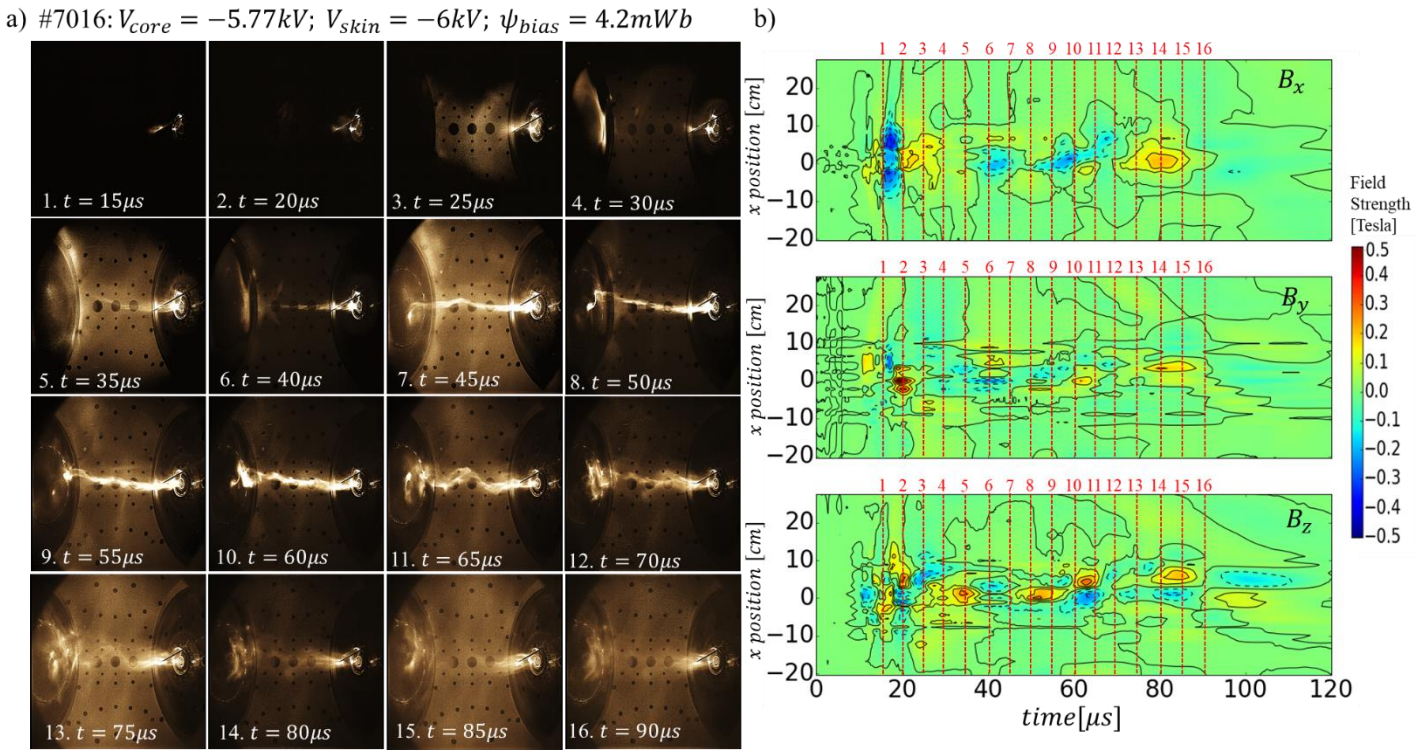


Figure 6.6. Magnetic field evolution for a more collimated shot #7016 using a single linear array 21cm in front of the gun electrodes shows simultaneous strong positive and negative axial fields which flip about $x = 0$. a) High speed images taken with the Kirana camera at 1 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system.

Following calibration of the remaining magnetic probe arrays (arrays 2 and 3), they were mounted radially inward from the top of the chamber at axial positions $z = 0.25$ m and 0.88 m, respectively, while magnetic probe array 1 was moved to the central midplane at 0.565 m. Because of the limited number of digitizer channels, we elected to focus on the high-resolution section of array 1 (clusters 1.2.1 – 1.2.24), while four central clusters (2.2.12 – 2.2.15 and 3.2.12 – 3.2.15) were measured for arrays 2 and 3, primarily for time of flight measurements. Figure 6.8 shows the evolution of the magnetic fields for magnetic probe array 1 for a representative shot #7089 during this phase of operation. Again, a primary feature is the regular reversal of the magnetic field in each orthogonal direction. This agrees with the observations taken closer to the electrode surface and suggest that the magnetic field structure is consistent along the length of the jet and is not well

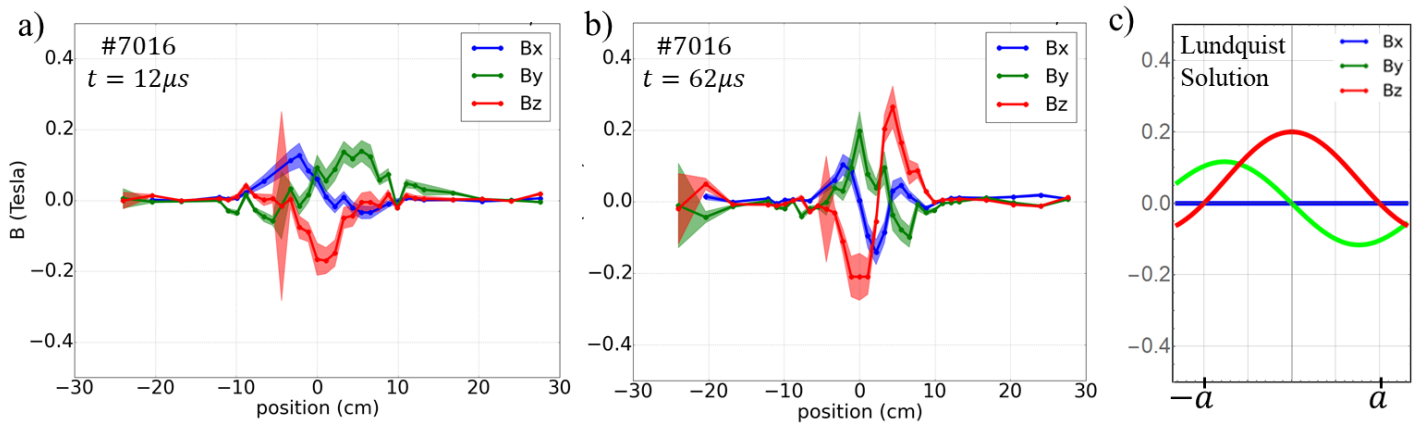


Figure 6.7. Magnetic field profiles at various times for shot #7016 a) Magnetic field profile for the stage I jet at $t = 12\ \mu s$. resembles a screw pinch. b) Magnetic field profile for the stage III jet with strong positive and negative axial fields not well described by a screw pinch. c) Negative helicity (i.e. $-\lambda$) Lundquist solution with a positive axial field. The profile is similar to the stage I jet but with the field components reversed.

described by an azimuthally symmetric screw pinch. It is also worth noting, based on the change in the ordinate scale length in figure 6.8b, that this jet appears to be more collimated in the central midplane than for previous measurements near the gun electrodes.

Because magnetic probe array 1 was mounted beyond the critical length of the stage I jet during this phase of operation, the initial screw pinch magnetic field profile is not observed. Instead, the initial magnetic probe signal reveals the propagation of two bow shocks with associated current sheets, as evidenced by the rapid reversal of B_x and B_y fields (Figure 6.9). The exact orientation of the fields varies from shot to shot, and between individual shocks; however, a double reversal appears to be a consistent feature. This suggests that each shock is formed independently, with the first shock resulting from the detached stage I jet as it propagates into the vacuum chamber (periods A and B in Figure 6.9 b, c) and the secondary shock formed by the secondary jet as it rams into the original shock (periods C and D in Figure 6.9 b,c). Evidence of this double shock is also observed in line integrated electron density measurements through the central midplane which exhibit a double peak at the start of the rise in density (Figure 4.16c,

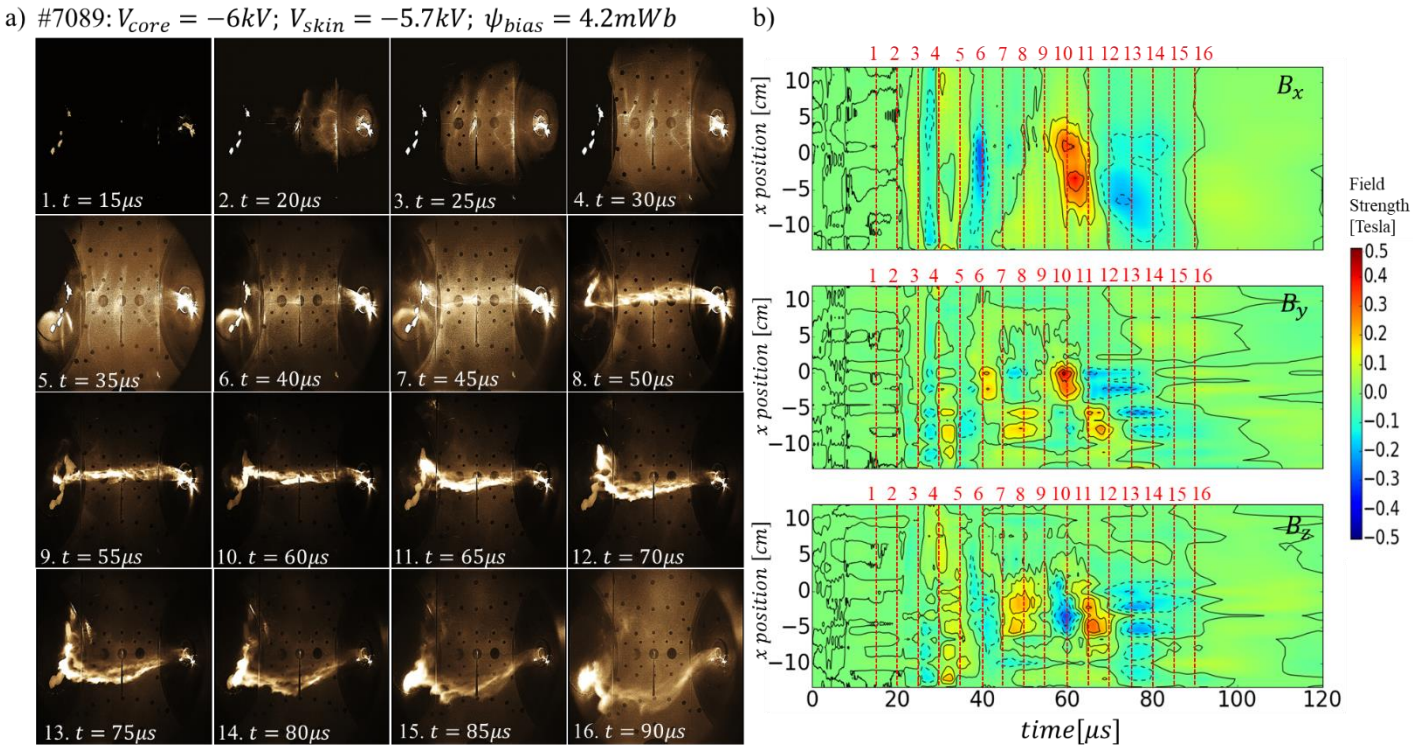


Figure 6.8. Magnetic field evolution for shot #7089 measured with the magnetic probe array in the central midplane also reveals a regular reversal of the magnetic field in each orthogonal direction. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system.

Figure 5.5c). Time of flight measurements using all three magnetic probe arrays indicate that the initial bow shock decelerates at $\sim 2 \times 10^6$ km/s^2 as it propagates across the chamber (Figure 6.10). The formation and rapid deceleration of the bow shock suggests a deceleration mechanism such as field line tension of the vacuum field.

In the final operational phase, magnetic probe array 1 was moved further away from the gun electrodes to $z = 0.74$ m while arrays 2 and 3 were rotated out from the central jet axis. Figures 6.11 and 6.12 shows the evolution of the magnetic field for two consecutive shots during this operation (#7175 and #7176 respectively). In both shots the results echo measurements obtained near the electrodes and in the central midplane, with a regular reversal of the magnetic field in all three orthogonal directions and a peak magnetic field strength of $\sim 0.4 - 0.5$ T.

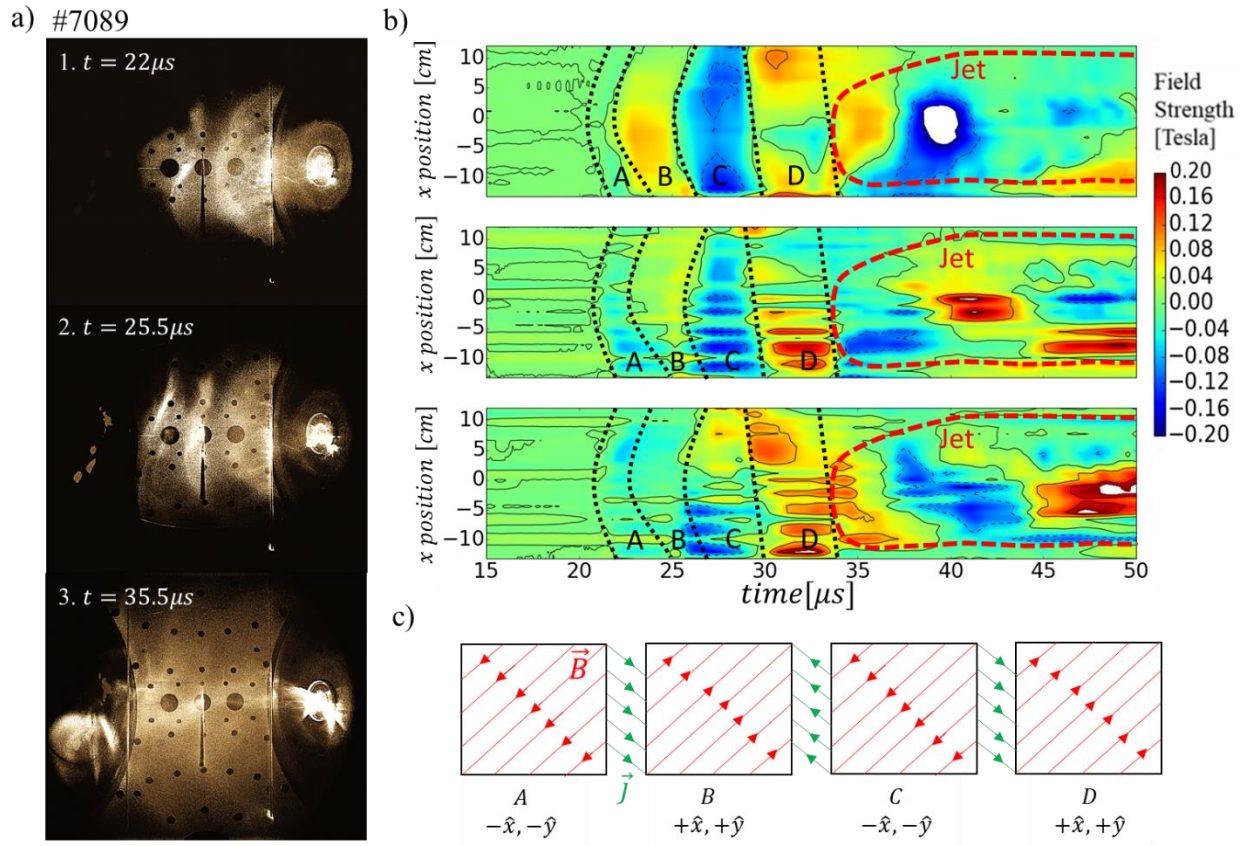


Figure 6.9. Detached plasmoid propagates as a bow shock followed by a secondary shock associated with the propagation of the secondary jet a) High speed camera images show propagation of initial detached plasmoid and the start of jet formation. b) Magnetic probe data in each orthogonal direction as the bow shock propagates past the array. Four distinct regions are highlighted corresponding to a reversal in B_x and B_y. c) Diagram of the magnetic field reversal with the associated direction of the current sheet between each region. A perfect reversal in magnetic field between regions B and C is not always observed.

Evidence of a bow shock is also observed but is more evident for shot #7176 with a broad, rapid reversal of B_x and B_y across all the measured clusters. Time of flight measurements using all three arrays corroborate bow shock deceleration on the order of $\sim 2 \times 10^6 \mathrm{km/s^2}$ (Figure 6.13). Later in time, for shot #7175, the magnetic probe array appears to pass slowly upwards through the central plasma column and the magnetic field contours bend in the $-\hat{x}$ direction with increasing time. For shot #7176 however, which appears to pass downwards through the central plasma column, the fields bend in the opposite $(+\hat{x})$ direction with increasing time. Because both shots are well collimated and do not oscillate significantly in the $\hat{x}$ direction over the transit time of the probe,

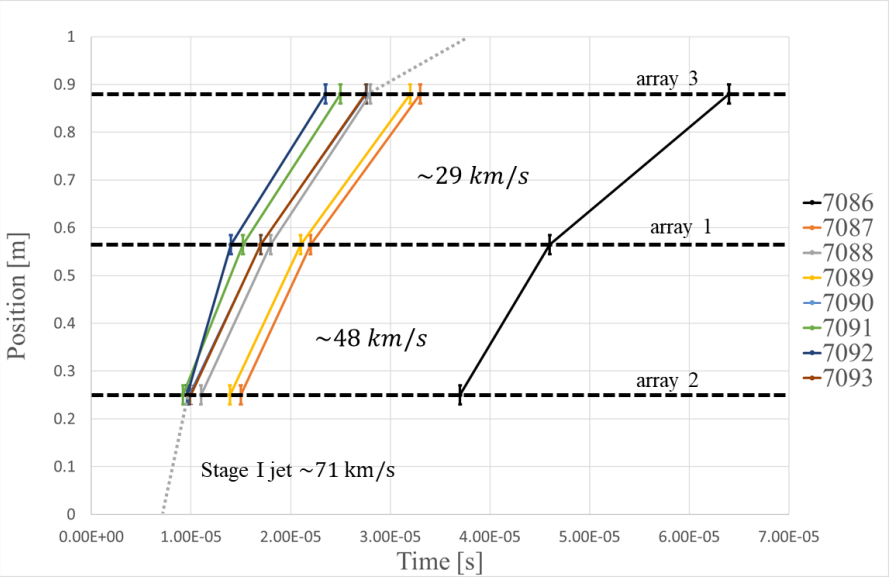


Figure 6.10. Time of flight measurements based on the arrival of the first magnetic field signal shows that the initial plasma bow shock decelerates at $\sim 2 \times 10^6$ km/s^2 as it propagates across the chamber. The formation and rapid deceleration of a bow shock suggest a deceleration mechanism such as field line tension of the vacuum field.

this suggests an observational selection effect related to the intrinsic magnetic field structure of the jet and the direction through which the probe is sampling.

6.3. Magnetic field cross section from a single linear Ḃ probe array

The magnetic field data presented in the previous section is particularly interesting in that it suggests that the magnetic field structure of the long stable jets is not reflective of a simple, azimuthally symmetric screw pinch. Unfortunately, with the number of digitizer channels limiting high resolution measurements to a single linear array it is challenging to infer what the 2D and 3D structure of the jet may be from such measurements. If local pressure data was available at each cluster position, the magnetic field cross section could potentially be recovered using a Grad-Shafranov solver in a similar manner to how cross sections of magnetic flux ropes in the solar wind have been reconstructed using a single spacecraft trajectory [98]. Without such pressure data; however, we use the motion of our single probe array through the plasma column to deduce properties of the magnetic field cross section for an individual shot.

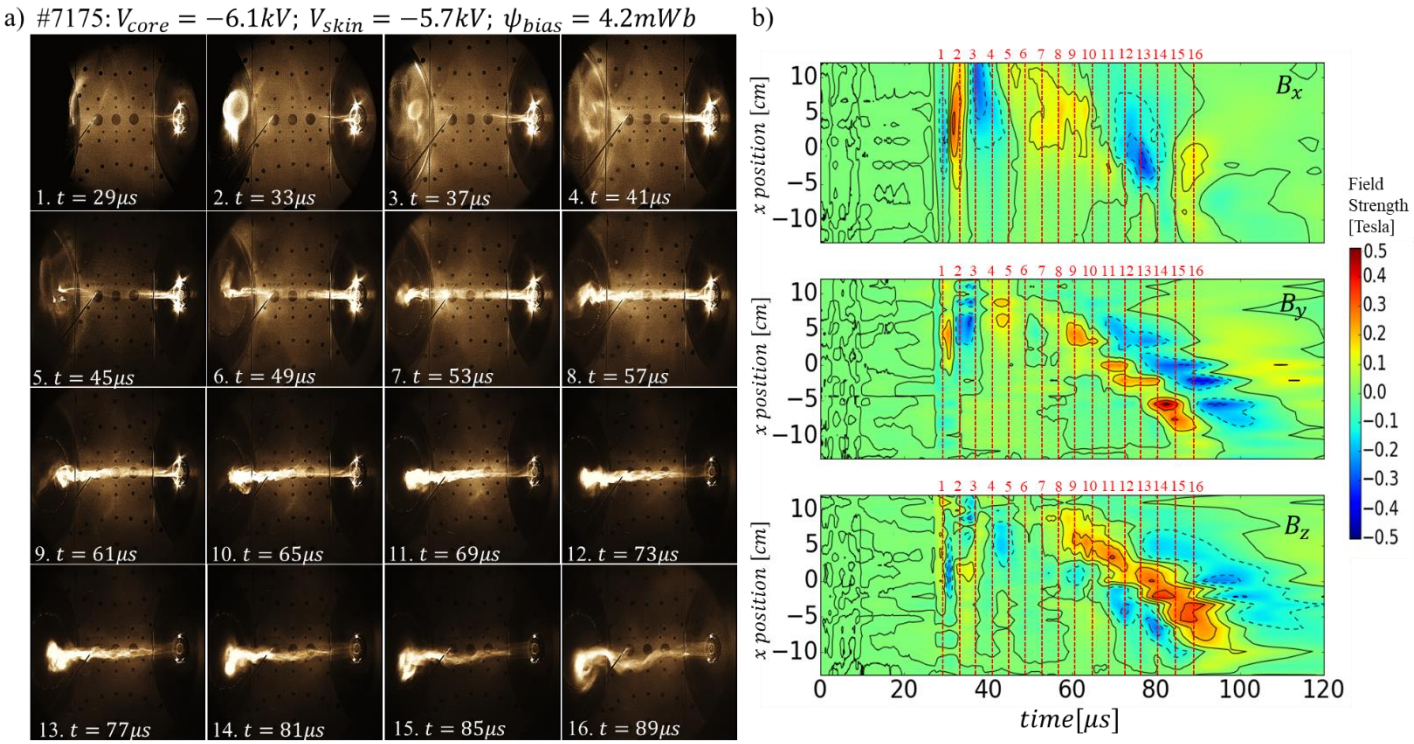


Figure 6.11. Magnetic field evolution for shot #7175 measured with a single probe array at $z = 0.74$ m. In this shot, the jet remained well centered on the probe for $> 40\mu s$ before traversing upwards through the column. A regular reversal of the field directions and a gradual bending of the magnetic fields in the $-\hat{x}$ direction suggest an observational selection effect possibly related to rotation of the magnetic field structure. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system.

The accuracy of this method is predicated on several assumptions. The first is that the magnetic probe array transit through the jet is nearly perpendicular to the jet axis (i.e. the jet is well aligned with the $\hat{z}$ axis). The second assumption is that the jet is axially uniform over the transit time of the probe (i.e. $\partial/\partial z \approx 0$) such that variations in the magnetic field in the axial direction are more gradual than those in the transverse direction as the jet propagates axially and the probe crosses transversely. Provided that these assumptions are met, a cross section can be reconstructed as shown in figure 6.14 by overlaying contour plots of the axial magnetic field as a function of time with vector plots of the azimuthal field components as a function of time. It is important to note that there are two possible scenarios for probe transits in the transverse direction,

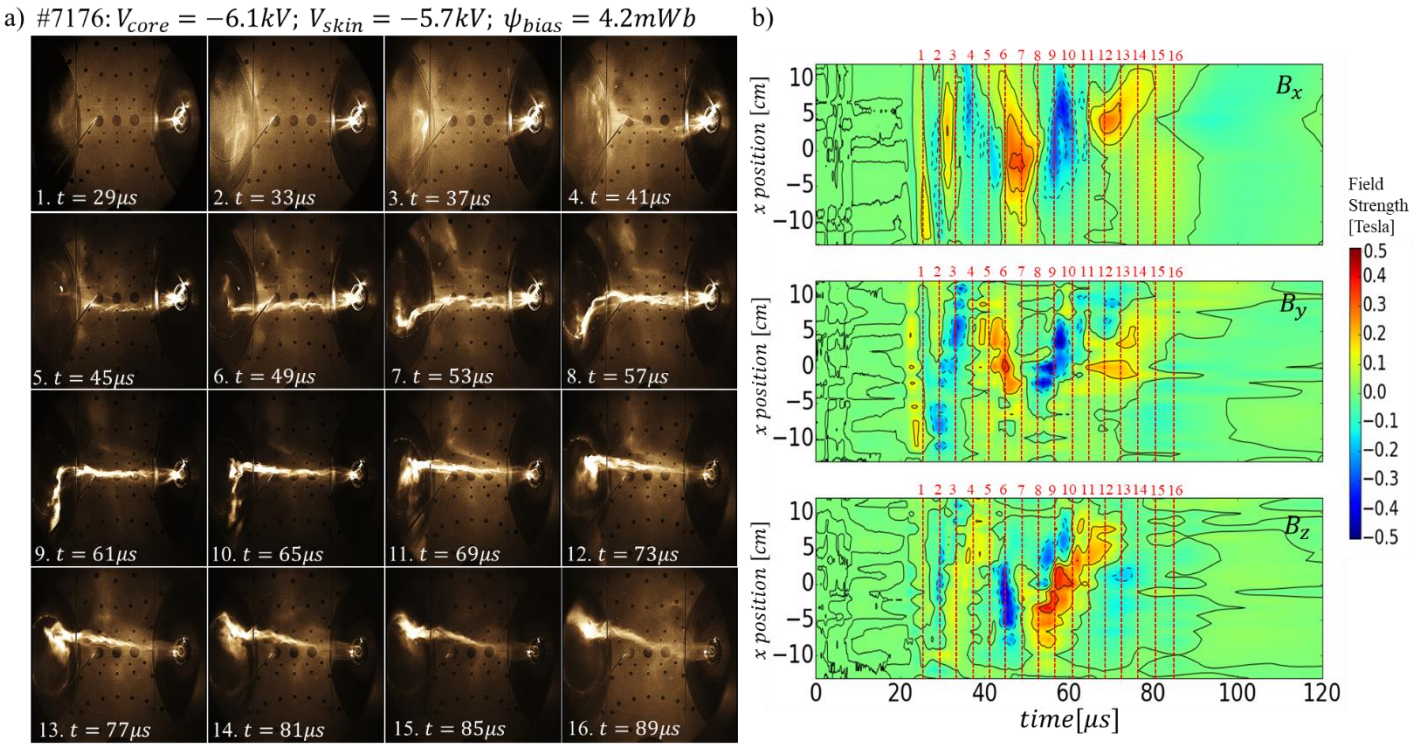


Figure 6.12. Magnetic field evolution for shot #7176 measured with a single probe array at $z = 0.74$ m. In this shot the probe was observed to traverse downward through the plasma column and a large regular reversal of the magnetic field was again observed. In this case, the fields appear to bend gradually in the $+\hat{x}$ direction. a) High speed images taken with the Kirana camera at 2 Mfps. b) Contour plots of the magnetic field evolution in each orthogonal chip direction. The $\hat{x}$ direction points towards the magnetic probe box mounted on the far side of the chamber in camera images, the $\hat{z}$ direction points out of the gun, and the $\hat{y}$ direction points downward to complete the right hand orthogonal coordinate system.

each requiring a different treatment to reproduce the appropriate cross section. For transits in which the probe crosses downward through the plasma column (i.e. $+\hat{y}$ direction) one must plot decreasing time on the ordinate axis while for transits upwards through the plasma column (i.e. $-\hat{y}$) one must plot with increasing time on the ordinate axis. Failure to plot the time in the correct direction will turn a circular vector field into a field approximating a saddle node bifurcation in phase space. For jets in which the transit of the probe is not in a single coherent direction (i.e. the jet oscillates up and down in front of the probe), one would expect regions where cross sections must be plotted with increasing time and other regions with decreasing time to obtain consistent results.

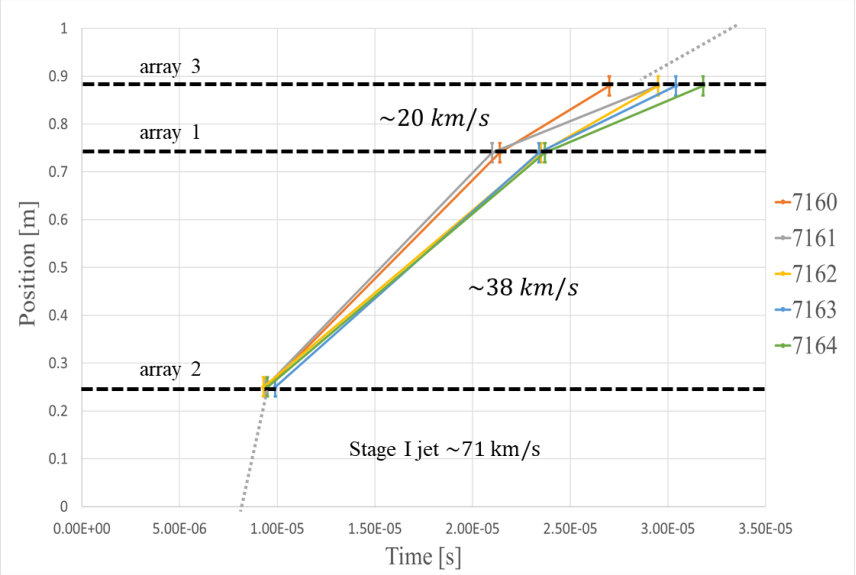


Figure 6.13. Time of flight measurements based on the arrival of the first magnetic field signal shows that the initial plasma bow shock decelerates at $\sim 2 \times 10^6$ km/s^2 as it propagates across the chamber. The formation and rapid deceleration of a bow shock suggest a deceleration mechanism such as field line tension of the vacuum field.

Figure 6.15 demonstrates the results of this analysis for shot #6999 where the magnetic probe array is seen to transit upwards through the central plasma column between $t \approx 25 - 70\mu s$ before reversing direction and transiting downward from $t \approx 70 - 110\mu s$. The results suggest that the probe traverses through multiple distinct magnetic flux tubes with opposite axial flux. For tubes with positive axial flux (Box 1 and 3 in Figure 6.15a,c) the azimuthal field is counter-clockwise while for tubes with negative axial flux (Box 2 and 4 in Figure 6.15a,c) the azimuthal field is clockwise. This corresponds to left handed helices arising from axial currents in the opposite direction of the axial magnetic fields (i.e. negative helicity, $-\lambda$). Such a negative helicity content within the jet is expected based on the orientation of the bias magnetic field ($+\hat{z}$) and the direction of current into the central electrodes ($-\hat{z}$).

If the cross section of each individual tube is approximately circular (i.e. the extent in the $\hat{x}$ direction is equal to the extent in the $\hat{y}$ direction) one can convert the time extent into a spatial extent in the $\hat{y}$ direction

$$y = \frac{\Delta x}{\Delta t}(t - t_{mid}) \tag{6.8}$$

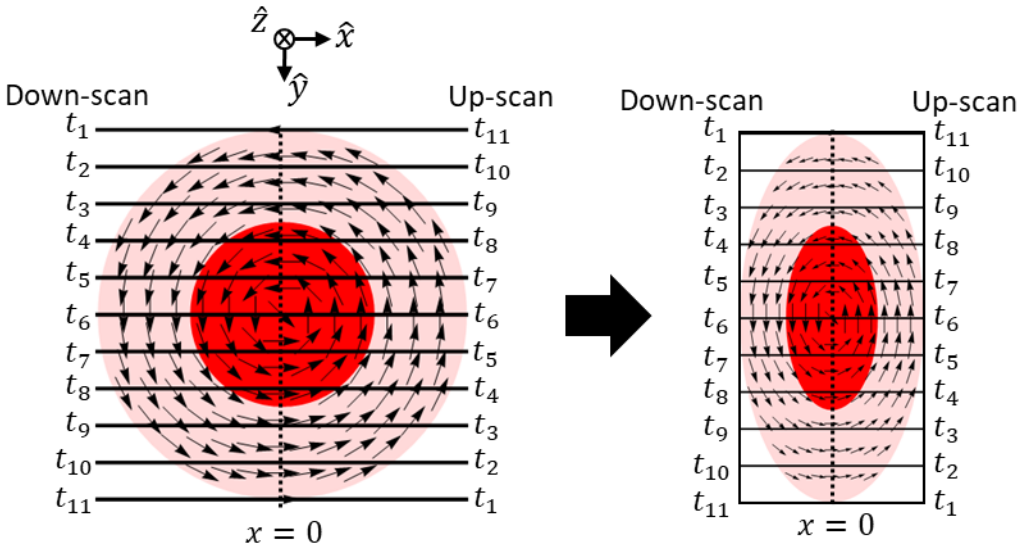


Figure 6.14. Method to reconstruct the cross section of a magnetic flux tube using a single magnetic probe array. The magnetic probe can traverse the tube in two directions (up-scan and down-scan) while measuring the magnetic field in all three orthogonal directions. One can therefore reconstruct the cross section of the flux tube (assuming axial uniformity during probe transit and alignment of the tube with the $\hat{z}$ direction) by overlaying contour plots of the axial field B_z with vector plots of the azimuthal field components B_x and B_y. To obtain the appropriate cross section, one must plot the ordinate with decreasing time values for down-scans, and increasing time values for up-scans.

where Δx is the diameter of the tube in the $\hat{x}$ direction, Δt is the diameter in the time axis and t_{mid} is the time at the midpoint of the cross section. With the cross sections now in purely terms of spatial dimensions, the axial current can be calculated

$$j_z = \frac{1}{\mu_0}\left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right) \tag{6.9}$$

as well as the $\lambda = \mu_0 j_z / B_z$ profile within each flux tube (Figure 6.15 b, d). To clean up the profiles, the raw data for each cross section is interpolated onto a 100×100 grid using a linear interpolation before calculating the spatial derivatives. The results confirm large antiparallel current densities on the order of $3 - 4 \times 10^7$ A/m^2 between $t = 25 - 50 \mu s$, which based on the cross-sectional area (~ 0.004 m^2) for each tube is in good agreement with the total gun current at

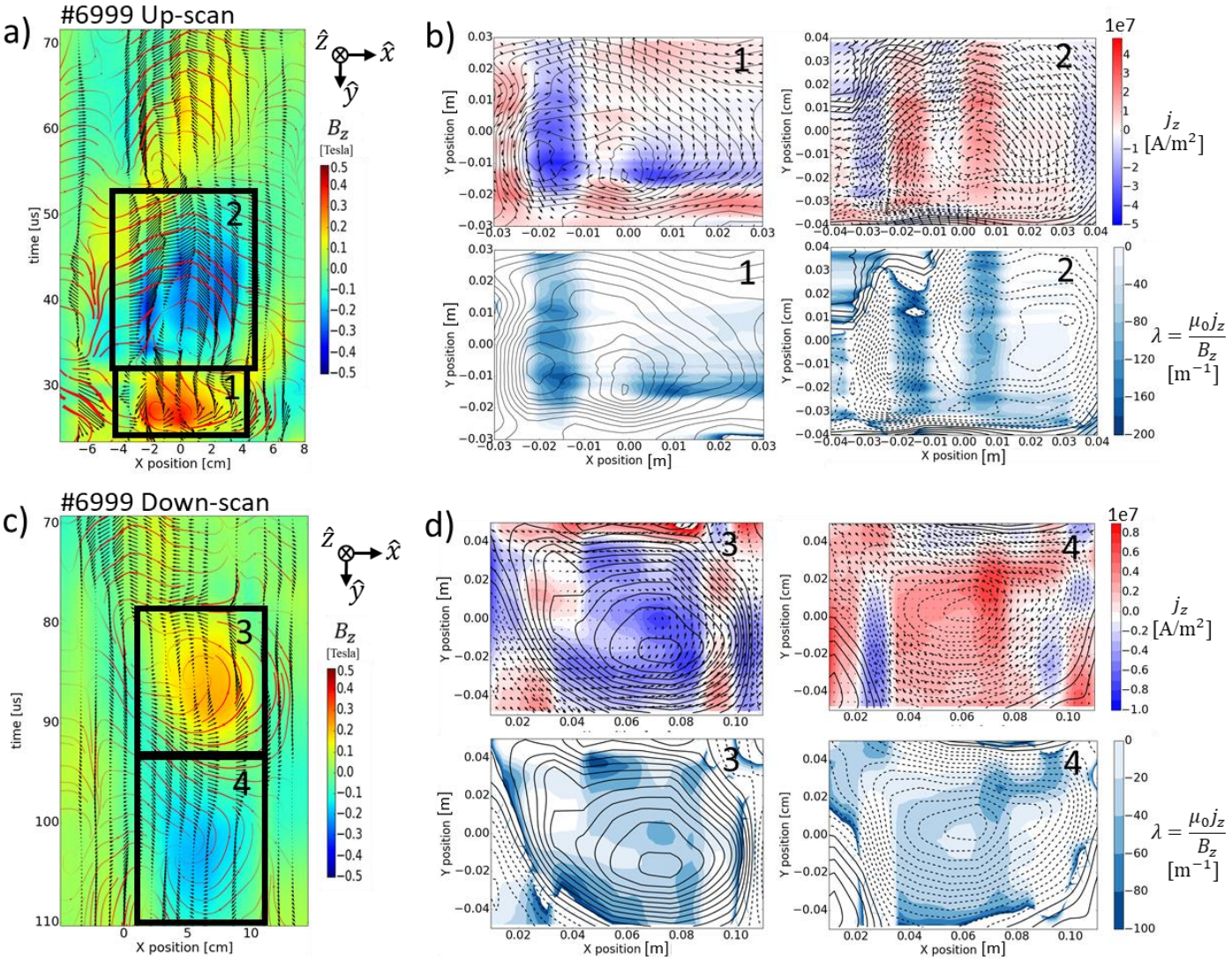


Figure 6.15. Cross section analysis for shot #6999 reveals opposing left handed flux tubes with large antiparallel currents and negative, non-uniform λ profiles. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y. b) Axial current and λ profiles overlaid on contours of axial magnetic field. c) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan of the probe array. d) Current density and λ profiles for flux tubes observed in the down-scan reconstruction.

this time which is between 150-170 kA. Reduced current densities on the order of 0.6×10^7 A/m^2 are observed later between 80-110 μs; however, based on the cross-sectional area of these tubes (~ 0.008m^2) the resulting total current is larger than the gun current which goes from 40 kA to -40 kA, crossing through zero at $t = 90\mu s$.

The λ profile for each tube is large and negative near the tube center but is not uniform throughout the cross section. Misalignment between the peak λ and the peak B_z may arise due to the jet not being axially uniform over the transit time of the probe or if the flux tubes are not perfectly aligned with the $\hat{z}$ axis. At $t = 25\mu s$ a double peaked $\lambda \approx -160$ m^{-1} overlaps well with the double peaked axial magnetic field. With a radius of $\sim 0.03m$ this corresponds to a normalized $\lambda a \approx -4.8$. At $t = 40\mu s$, the agreement between the λ profile and the magnetic field contours is not as good, but values of $\lambda \approx -120$ m^{-1} and a radius of ~ 0.04 m results in a consistent normalized $\lambda a \approx -4.8$. Towards the end of the discharge between $t = 80 - 100\mu s$, peak λ values drop below ~ 60 [m^{-1}] while the radius increases to $\sim 0.05cm$ for a normalized $\lambda a \sim 3$.

Figure 6.16 shows the same analysis for magnetic probe array 1 in the central midplane for shot #7089. In this shot the probe traverses downwards through the jet between $t = 45 - 54\mu s$ before the jet oscillates downwards and the probe crosses upwards through the well collimated central plasma column between $t = 54 - 70\mu s$. The reconstructed cross sections again reveal left handed helices with antiparallel current densities on the order of $2 - 3 \times 10^7$ A/m^2. For the first flux tube (Figure 6.16 a,b), the peak λ is ~ -160 m^{-1} with a radius of ~ 0.04 m resulting in a normalized $\lambda a \approx 6.4$. For the second two flux tubes (Figure 6.16 c,d), the observed λ profile is relatively uniform at $\lambda \approx 120$ m^{-1} and overlaps well with the magnetic field contours. Presuming that both positive and negative axial flux tubes comprise the core jet and that the jet has a circular cross section, the approximate jet radius is ≈ 0.05 m resulting in a normalized $\lambda a \approx 6$.

Up to this point, the results of the cross-sectional analysis indicate that the two-dimensional structure of the jet consists of at least two magnetic flux tubes with opposite axial fluxes and current densities. Conducting the same analysis for shot #7175 however, provides additional information regarding the three-dimensional structure of the jet. For this shot, the magnetic probe array remained well centered for $> 40\mu s$. Despite being well centered, the reconstruction analysis in both up-scan and down-scan directions reveals transits of magnetic flux tubes across the magnetic probe array (Figure 6.17a). At $t \approx 65\mu s$ the probe traverses upwards through the positive axial flux tube at $x \approx 5$ cm and downward through the negative axial flux tube at $x \approx -5\ cm$ (black boxes in figure 6.17a). At $t \approx 85$. At $t \approx 85\mu s$ this process is reversed as the probe traverses downwards through the positive axial flux tube at $x \approx -5$ cm and upwards through the negative flux tube at $x \approx 5$ cm (red boxes in figure 6.17 a). These results suggest that the positive and negative flux tubes are rotating around a common central axis with a frequency of $\sim 25\ kHz$ in

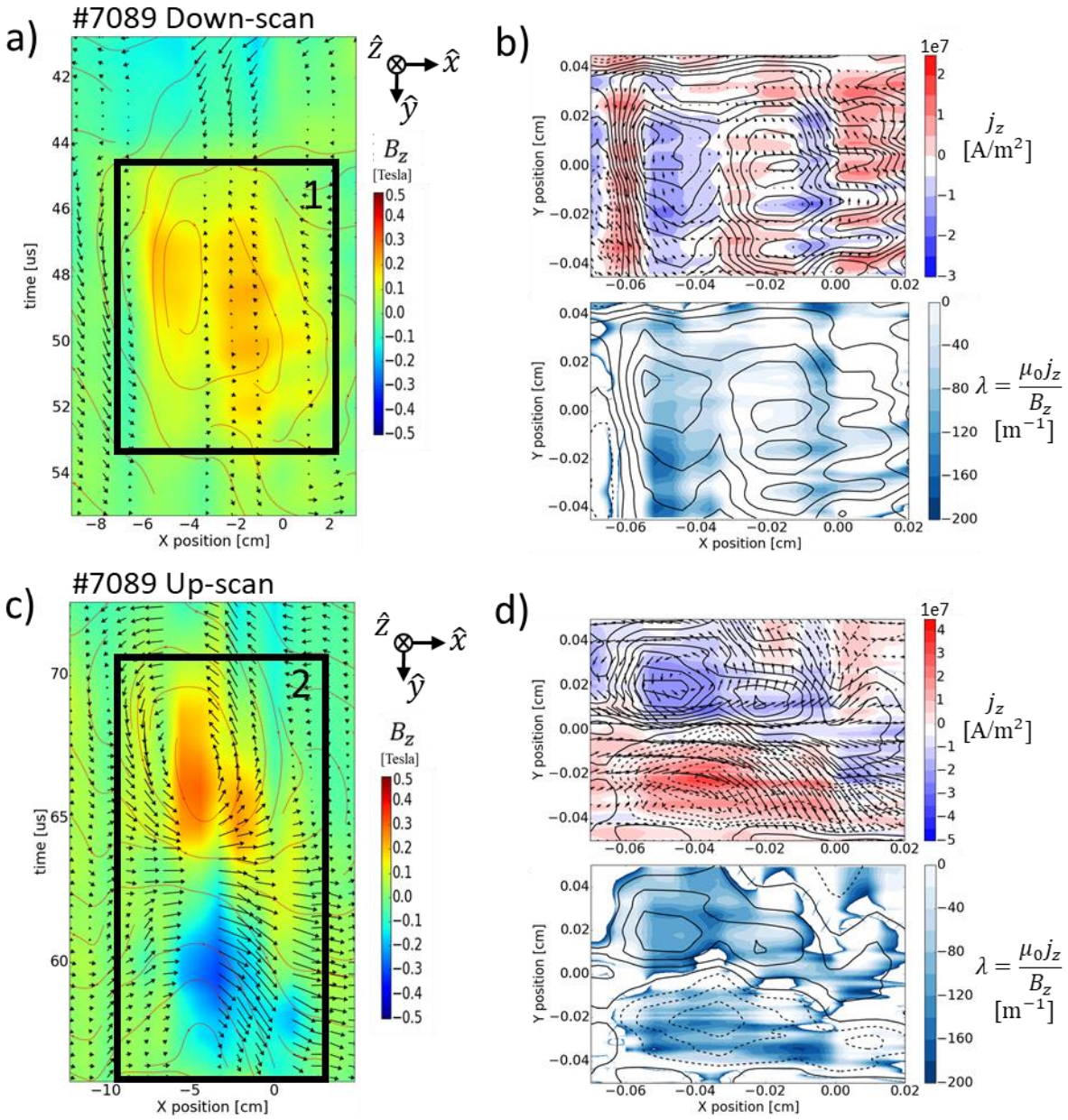


Figure 6.16. Cross section analysis for shot #7089 with the probe array in the central midplane also reveals left handed flux tubes with large antiparallel currents and a negative λ. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan of the probe array. b) Current profile and λ profiles overlaid on contours of axial magnetic field. c) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for an upward scan of the probe array. d) Current and λ profiles for up-scan. The λ profile during this scan is relatively flat in both tubes with $\lambda \approx 140$ m^{-1}.

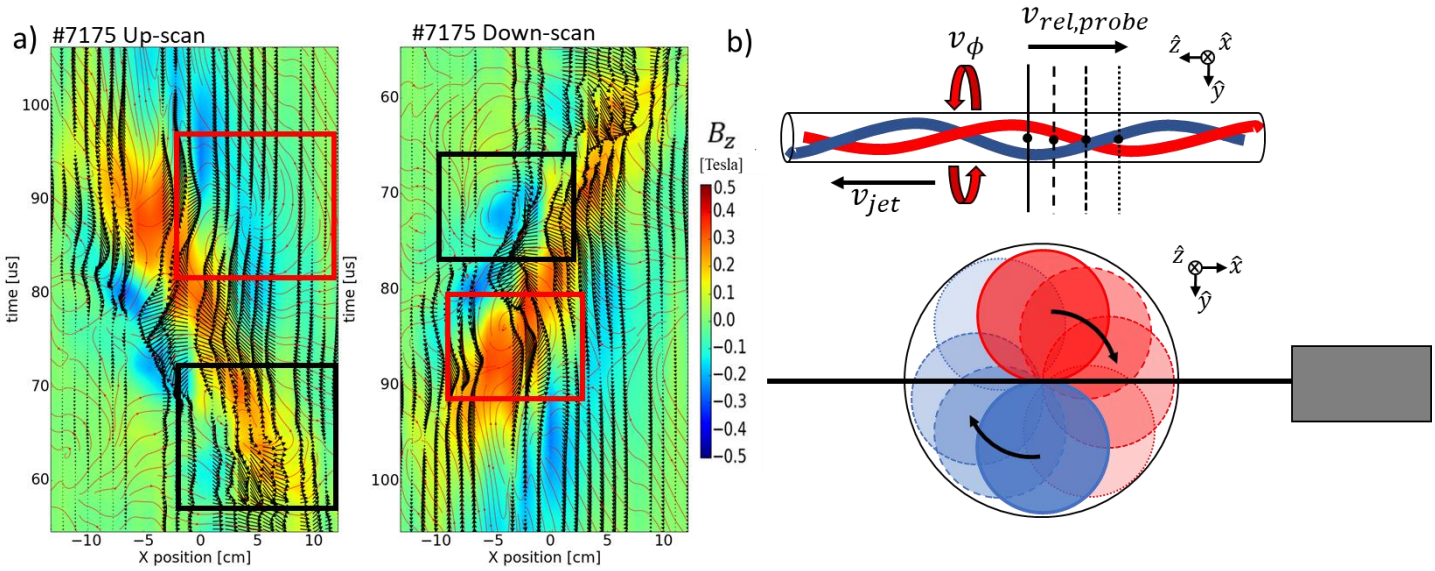


Figure 6.17. Cross section analysis for shot #7175, which remained well centered on the probe array for $> 40\mu s$, suggests helical and/or rotating flux tubes. a) At $t \approx 65 - 70\mu s$ the magnetic probe array traverses upward through a positive axial flux tube and downward through a negative axial flux tube (black boxes). Later at $t \approx 85 - 90\mu s$ the probe array traverses downward through a positive axial flux tube and upward through a negative axial flux tube (red boxes). b) Interpretation of the results from the point of view of two helically intertwined flux tubes with opposite axial fluxes suggests that the helix rotates at ~25 kHz or equivalently the magnetic probe array translates axially through the helix resulting in an apparent rotation in the magnetic probe data (or both).

the $+\hat{\phi}$ direction. This corresponds to an azimuthal velocity of the flux tube centers of ~ 8 km/s which agrees well with the skin azimuthal velocity fit parameter for the double shell ion doppler spectroscopy model (section 5.4.1). An alternative (but non- contradictory) interpretation presumes that the jet structure is helical, in which case axial translation of the magnetic probe array due to the relative motion of the jet can also result in an apparent rotation (Figure 6.17 b). Based on the observed rotation frequency, for an axial propagation velocity on the order of $25 - 50$ km/s, the axial wavelength would need to be between $1 - 2$ m. Camera images however (Figure 6.11a), indicate that the axial wavelength of the helix is < 0.5 m, which suggests that axial translation alone cannot account for the observed rotation frequency.

6.4. Comparison to helical Taylor State

In conjunction with high speed camera images, the results of the cross-sectional analysis suggest that the magnetic field structure of the jet consists of at least two helically intertwined magnetic flux tubes with opposite axial fluxes and antiparallel current densities, each with a normalized $|\lambda a| > 3$. Such a state ($m = 1$ Taylor state) is theoretically known to exist for plasmas bounded by a cylindrical flux conserver with aspect ratios >1.67 [13, 14] (see section 1.3.3) and has been observed experimentally [17, 99] for the aspect ratios ≥ 3, but is not predicted in unbounded volumes. Despite this, the remarkable similarity between the observed magnetic field data and what might be expected of a helically relaxed Taylor state warrants further investigation. To conduct this investigation, we solve for the magnetic field of a helical Taylor state with an aspect ratio of 25 using the method outlined in section 1.3.3 (Figure 6.18). This solution has a normalized eigenvalue $\lambda a = -3.119$ (negative to reflect the negative helicity content observed in our jets) which is approximately equivalent to the infinite cylinder negative helicity solution $\lambda a = -3.11$. Magnetic field traces are obtained by assuming a single linear array traverses through the structure with a prescribed trajectory. To simplify our comparison, the model is assumed to be static in time (i.e. non-rotating, non-oscillating) such that the probe motion itself is responsible for changes to the observed magnetic field. Because the magnetic field strength decreases as one approaches the ends of the model at $z = 0$ and $z = L$, we consider transits through the central region only. To qualitatively compare the model traces to our experimental data, we replicate the magnetic field cross section plots described in the previous section with the time axis replaced by axial position of the probe within the model. Reversing the direction in which the axial position is plotted is equivalent to flipping the time axis and can therefore account for up-scans, and down-scans through the jet in the transverse direction.

Figures 6.19 – 6.21 show the results of this comparison for three experimental shots. In the first shot, #7150 (figure 6.19) high speed video shows that the magnetic probe array traverses downwards ($+\hat{y}$) through the central plasma column between $t \approx 35 - 70\mu s$ (see figure 4.16a). To replicate this transit through the Taylor state model, the *y* coordinate of the linear probe array is selected to vary with the axial position as $y_{probe} = -2\text{a}\sin(0.08z/a + 2)$ such that the probe traverses the model in the $+\hat{y}$ direction between between $z/a \approx 8 - 20$. Because this is an downward scan, time and axial position are ploted in decreasing order on the ordinate axis (figure 6.19 a,b); however, as can be seen between $z/a = 12.5 - 14.5$, flux tubes (in this case

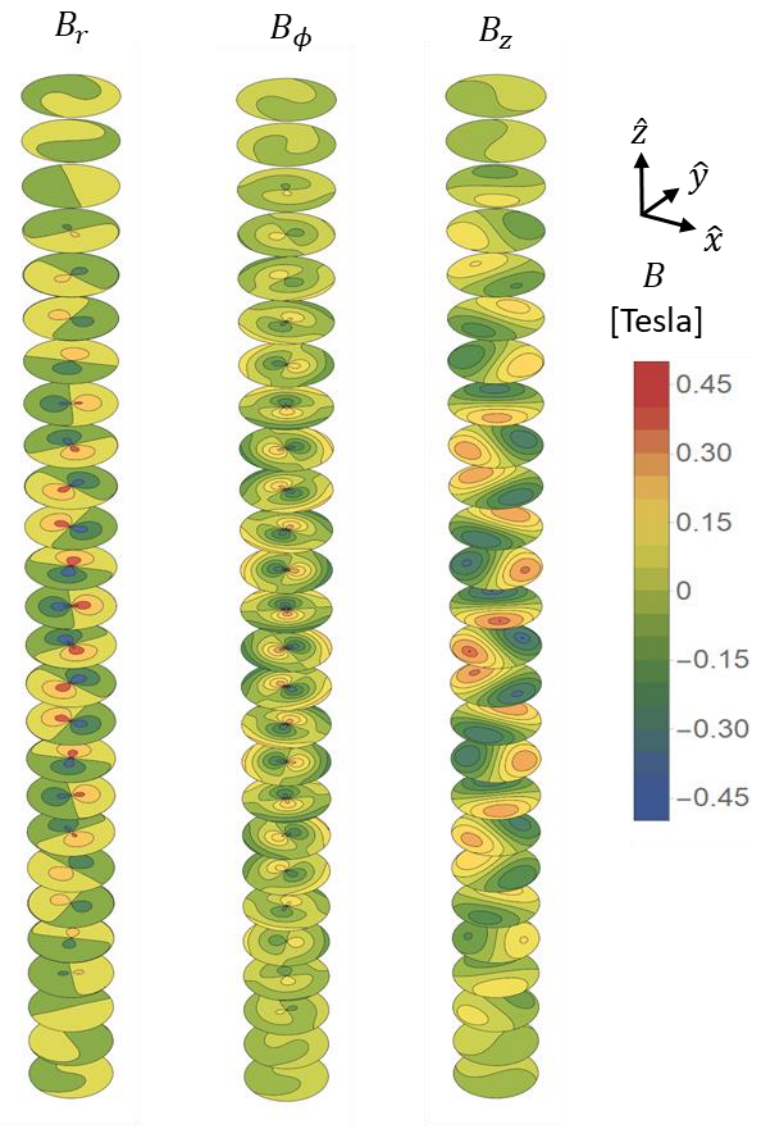


Figure 6.18. Helical $m = 1$ Taylor state model used to compare with experimental data. The model has an aspect ratio of 25 with $\lambda a = 3.119$. The model does not rotate or oscillate in time such that the hypothetical probe trajectory is solely responsible for changes in the observed field.

the positive flux tube) can cross the magnetic probe array in the same direction as the probe scan due to helical winding in the $+\hat{\phi}$ direction with increasing axial position (or azimuthal rotation in the $+\hat{\phi}$ direction for the experimental data). Comparison of the experimental and model field traces reveals a remarkable agreement, with specific regions for comparison circled in red and black respectively. Early in time (axial position) as the probe array first enters the plasma column, the negative flux tube sweeps along the array in the $+\hat{x}$ direction (circle 1). As the probe continues

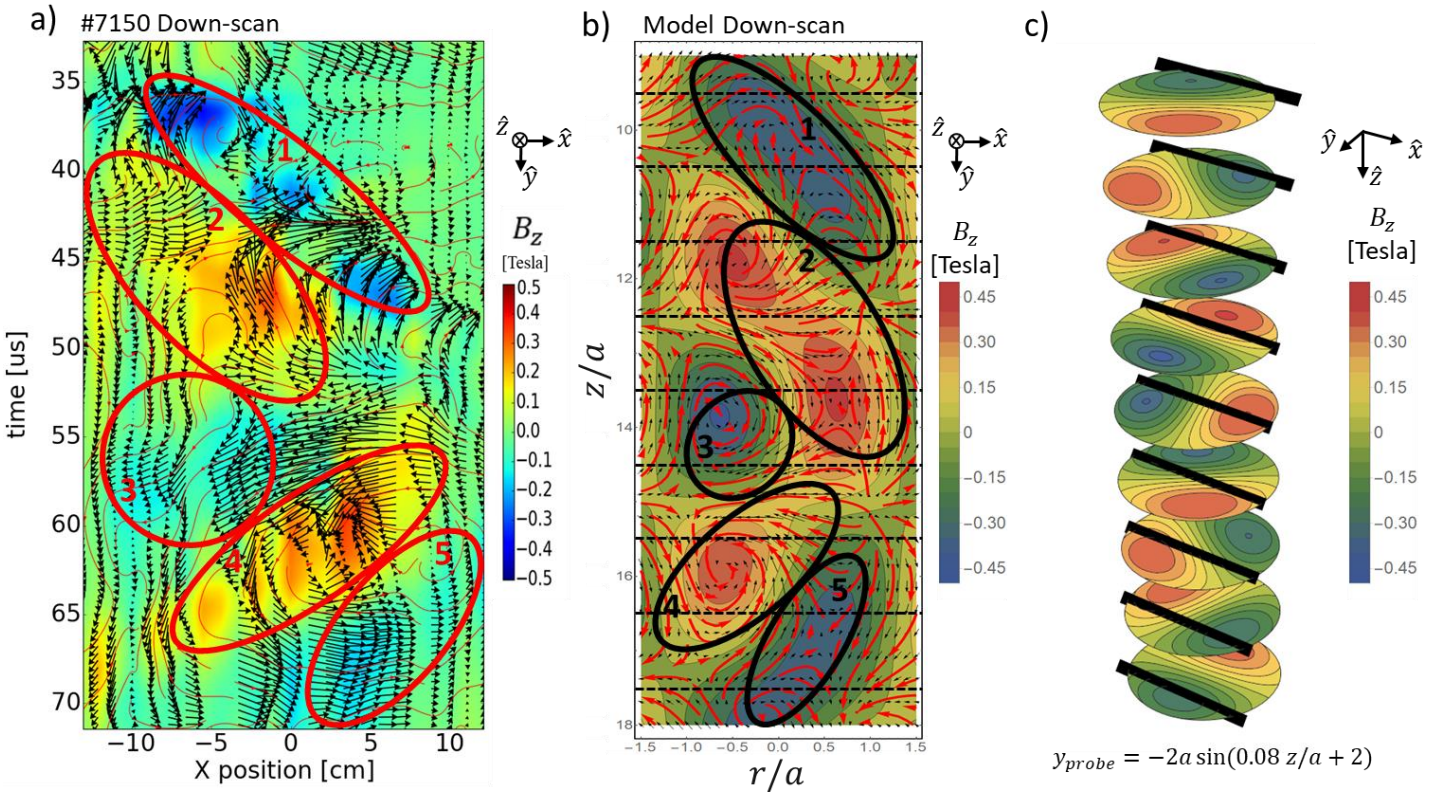


Figure 6.19. Comparison of experimental magnetic field data from shot #7150 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for a downward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model.

its transit, the positive tube crosses the array in the $-\hat{y}$ direction, sweeps along in the $+\hat{x}$ direction before crossing the probe again in the $+\hat{y}$ direction (circle 2). The experimental data appears to just catch the end of this first transit, as evidenced in the top left corner of circle 2, but appears to catch the full second transit in the direction of the probe scan, which results in a vector field resembling a saddle node bifurcation. In the experimental data, circle 2 is offset in the $-\hat{x}$ direction compared to model traces, which may result from slight oscillation or bowing of the jet. Concurrent with the second transit of the postive flux tube, the negative flux tube traverses the probes in the $-\hat{y}$ direction (circle 3). This feature is also observed in the experimental data, but is more diffuse

and only appears to catch a portion of this transit since the full clockwise vector field is not observed. Once the probe has crossed the center of the jet, the positive flux tube makes one more transit across the probe in the $-\hat{y}$ direction (circle 4) before the negative flux tube sweeps along the probe in the $-\hat{x}$ direction as it exits the jet (circle 5).

Figure 6.20 presents the same analysis for shot #7153. In this shot, the probe array is observed to transit upwards ($-\hat{y}$) through the plasma column between $t \approx 35 - 70\mu s$. To replicate this transit through the model, the y coordinate of the hyptothecial array is chosen to varry as $y_{probe} = 2a\sin(0.1z/a + 1.5)$ such that the probe traverses the model in the $-\hat{y}$ direction between $z/a \approx 11 - 22$. In this case, because the probe scans upwards through the jet, time and axial position are ploted in increasing order on the ordinate axis (Figure 6.20 a,b). A detailed comparison between the experimental and model field traces once again reveals an uncanny resemblance, with regions of similarity circled in red and black respectively. As the probe first enters the plasma column, the negative flux tube rotates in the $+\phi$ direction with increasing axial position such that the probe does not fully traverse the tube (circle 1). The positive tube however, which is also rotating in the $+\phi$ traverses the probe in the $+\hat{y}$ direction resulting in a counter clockwise vector field observed in both the experimental and model traces (circle 2). This positive tube continues its rotation and crosses the probe again in the opposite direction (circle 3) resulting in a vector field resembling a saddle node bifurcation. Concurrently, the negative flux tube traverses the probe in the $+\hat{y}$ direction resulting in a clockwise vector field (circle 4). After crossing the center of the jet, the positive flux tube once again traverses across the probe in the $+\hat{y}$ direction resulting in counter clockwise vector field (circle 5), followed by the negative flux tube which does not complete a full transit before the probe leaves the plasma column.

A final comparison was conducted for shot #7175 (Figure 6.21), for which the probe array remained well centered on the jet between $t \approx 55 - 90\mu s$. If the probe array was perfectly centered through the helical Taylor state model however, one would expect simultaneous, alternating positive and negative axial field signatures centered around $x = 0$. Instead the experimental magnetic field traces are elongated across the central axis, with a positive flux component bordered by two negative flux regions. This suggests that the probe array is not perfectly centered, and thus, for the hypothetical probe transit through the model, the y coordinate of the probe was fixed at $0.4a$ while the probe traversed axially from $z/a = 12$ to $z/a = 19$. Once again, the experimentally measured magnetic field traces and the reconstructed model traces bear

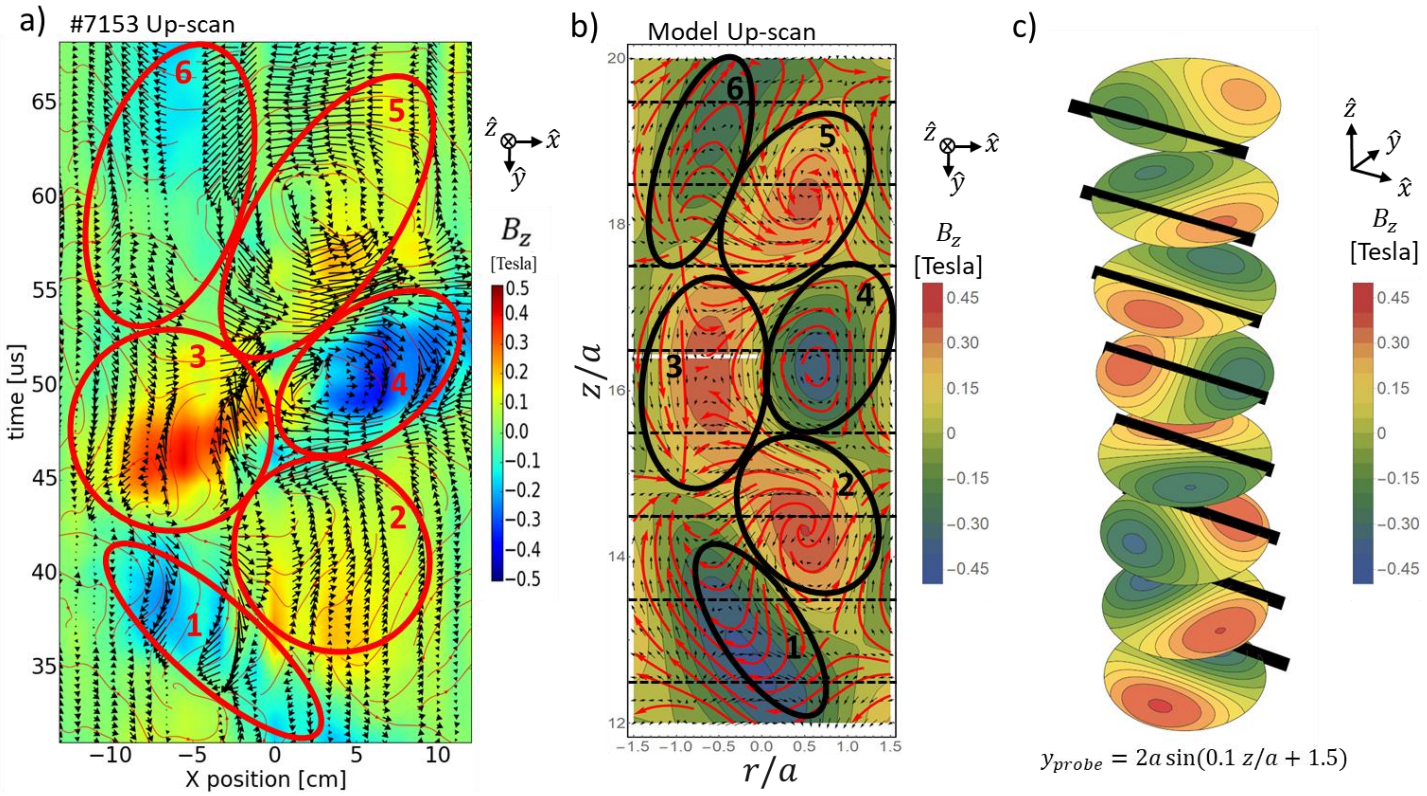


Figure 6.20. Comparison of experimental magnetic field data from shot #7153 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y for an upward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model.

a striking resemblance. With the hypothetical magnetic probe array fixed in y, a negative flux tube initially sweeps along the probe in the $-\hat{x}$ direction due to the $+\hat{\phi}$ twist in the axial direction (circle 1). The resulting vector field is nearly identical to that of the experimentally measured trace. As the hypothetical probe continues in the axial direction, the positive flux tube performs the same sweep with the centroid of the tube crossing the probe in the $+\hat{y}$ direction at $z/a \approx 14.5$ resulting in a counter clockwise vector field that is mirrored in the experimental data (bottom right of circle 2). The positive tube continues to rotate in the $+\hat{\phi}$ direction as the probe translates axially and the

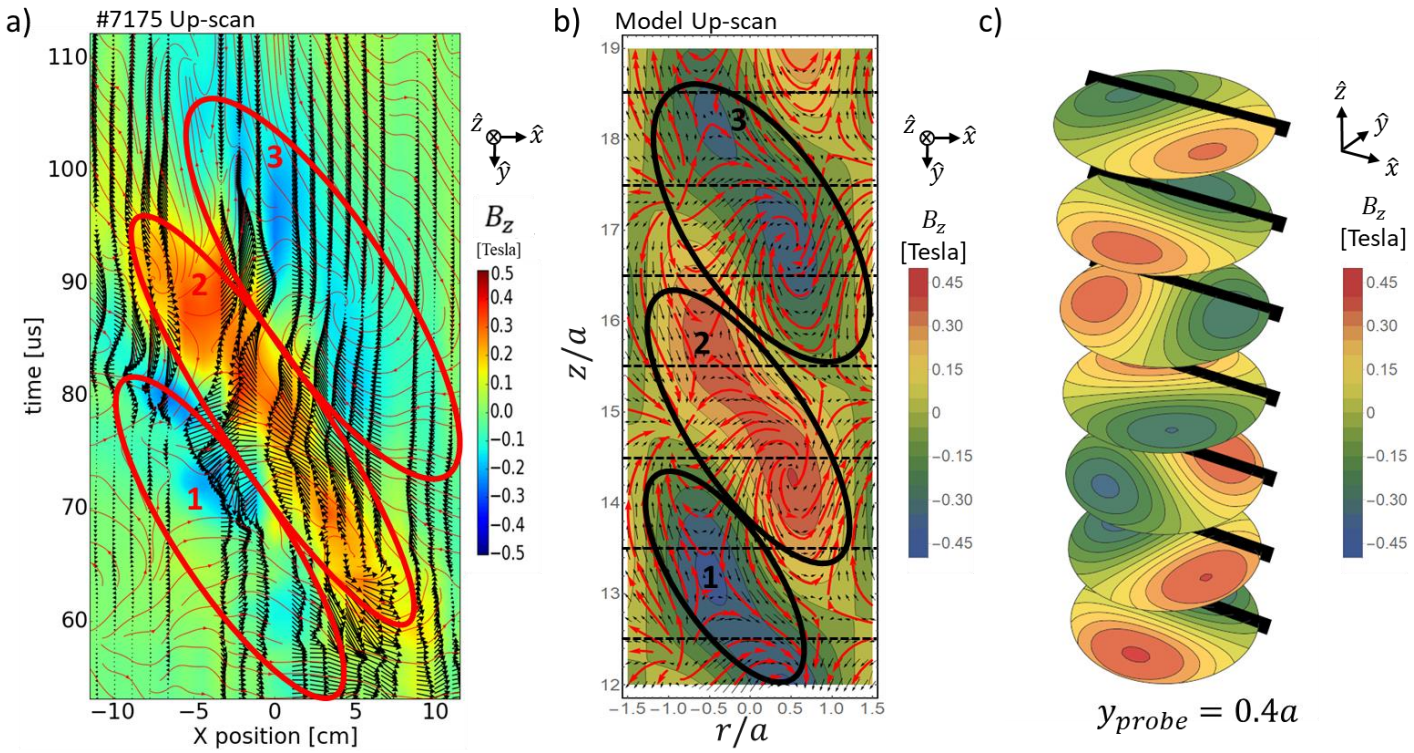


Figure 6.21. Comparison of experimental magnetic field data from shot #7175 to that of a hypothetical transit through an $m = 1$ Taylor state with an aspect ratio $L/a = 25$. a) Contour plot of the experimentally measured axial magnetic field B_z overlaid with a vector plot of the azimuthal field components B_x and B_y presuming an upward scan through the plasma column. Regions of similarity have been circled. b) Corresponding contour plot resulting from trajectory through the magnetic field model. Dashed black lines correspond to the axial position of slices in panel c. c) Isosurfaces of the magnetic field model at various axial positions. Black lines indicate the position and orientation of the hypothetical linear probe array as it traverses through the model.

centroid crosses the probe again in the -$\hat{y}$ direction at $z/a \approx 15.5$ resulting in a vector field resembling a saddle node bifurcation (top left of circle 2). This process is then repeated once more for the negative flux tube, resulting in a counter clockwise vector field in the bottom right of circle 3 followed by a saddle node in the top left.

6.5. Discussion

This chapter presents the results obtained using three custom magnetic probe arrays on the MoCHI.LabJet experiment. Observations were limited by the number of digitizer channels available such that most of the measurements presented are from a single, high-resolution section of magnetic probe array 1. Measurements with the remaining probes were used primarily for time of flight analysis. With the magnetic probe array located near the LabJet gun electrodes, the initial magnetic field structure of the stage I jet was observed to be well described by an azimuthally symmetric screw pinch similar to a Lundquist flux rope. However, the orientation and strength of the magnetic fields in this stage I jet were opposite and weaker than the predicted based on the orientation of the bias magnetic field, propagation velocity of the jet, and total gun current. It is hypothesized that a large diamagnetic current due to large pressure gradient between the stage I jet and its surroundings could be responsible for these observations. Following the stage I jet, the formation of the stage II jet was characterized by rapid oscillation of the magnetic field signals which then entered a relatively quiescent period as the jet transitioned into the stage III, long, stable jet. Magnetic field traces from stages II and III suggested strong positive and negative axial flux components within the core of the jet which is inconsistent with an azimuthally symmetric screw pinch. These observations were confirmed in the central midplane where strong positive and negative axial flux components were observed as the central plasma column traversed the magnetic probe array. With the array beyond the critical length of the stage I jet during this phase of operation, the initial magnetic field signal was found to be indicative of a double bow shock with multiple associated current sheets. The first bow shock is thought to form as the stage I jet detaches and propagates across the chamber while the second shock is likely formed by the stage II jet as it collides with this initial shock. Time of flight measurements using all three magnetic probe arrays at various axial locations reveal that the initial bow shock decelerates rapidly at $\sim 2 \times 10^6$ km/s^2 suggesting a deceleration mechanism such as field line tension of the vacuum field. Similar results were also observed with magnetic probe array 1 moved towards the far end of the chamber at $z = 0.74$ m.

Despite high resolution measurements limited to a single linear array, it was determined that two-dimensional information of the magnetic field structure could be obtained by exploiting the natural oscillation of the jet transverse to the magnetic probe array. The resulting cross-sectional analysis revealed that the positive and negative axial flux components made up distinct

left-handed flux tubes with large antiparallel current densities and normalized $\lambda a < -3$. Such negative helicity (i.e. $-\lambda$) is expected due to the negative magnetic helicity injection provided by the LabJet gun with the bias field in the $+\hat{z}$ direction and antiparallel current into the gun electrodes. The same analysis for non-oscillating jets also hinted at the three-dimensional structure, suggesting that the magnetic flux tubes are wound helically around each other in a right-handed manner and/or rotate about a common axis in the $+\hat{\phi}$ direction. To test this hypothesis, experimental magnetic field traces were compared to hypothetical probe transits through a helically relaxed $m = 1$ Taylor state with an aspect ratio of $25{:}\,1$ corresponding to a $\lambda a = -3.119$. The results of the comparison are remarkable in their similarity, suggesting that the magnetic field structure of the stage III jet is in qualitative agreement with a helically relaxed Taylor state despite the absence of nearby flux conserving walls, the presence of strong axial and azimuthal flows, and a substantial plasma thermal pressure (i.e. $\beta \approx 1$). These elements may contribute to the lack of full agreement with a minimum energy Taylor state, such as a non-uniform λ profile and $|\lambda a| > 3.11$, and could potentially be accounted for by a more generalized relaxation process based on the conservation of canonical helicity (see chapter 2) or by the fact that the jet is being driven throughout the discharge and does not have time to fully relax.

Chapter 7
CONCLUSION

This thesis presented the motivation, foundational theory, design, and initial results of a novel pulse power plasma experiment designed to simulate astrophysical jets in a laboratory. The experiment is based around a planar triple electrode plasma gun that mimics an accretion disk threaded by a poloidal magnetic field. The two inner electrodes can be biased independently with respect to the outer electrode to control the radial electric field profile which, when crossed with the poloidal vacuum field, approximates the shear rotation profile in an accretion disk. Azimuthally symmetric and continuous gas injection slits allow for free rotation of the jet foot-points while control of the current profile provides some degree of control on the axial flow shear [33]. The primary goal of the project is improving our understanding of the interaction between flows, magnetic fields, and jet stability from the point of view of canonical flux tubes and canonical helicity.

Chapter 2 presents an overview of a canonical field theory which provides a unified framework for the dynamics of single-particle, kinetic, and fluid plasma models and is derived directly from a Lagrangian-Hamiltonian formalism [48]. The framework retrieves a canonical equation of motion which is valid across all regimes and reveals a more fundamental topological constraint than magnetic or fluid kinetic helicity. Because of the isomorphism between the governing equations and Maxwell's equations, the model suggests that techniques borrowed from the theory of electromagnetism are suitable for the analysis of plasma dynamics and relaxation, while a generalized frozen-in condition extends intuition developed in the ideal MHD regime across all plasma regimes, suggesting the existence of generalized non-Taylor equilibria with strong shear flows and high β [47, 48, 51, 54].

In Chapter 3, this theory is used to examine the morphology and topology of generalized, canonical flux tubes for increasing electrical currents, different radial current profiles, different electron Mach numbers, and a fixed, flared, axisymmetric magnetic geometry. These calculations are indented to provide an intuitive, kinematic view of the

evolution of jets with strong flows without resorting to costly fully self-consistent dynamical calculations. The results reveal the dependence of canonical fields on the underlying magnetic and flow components and show that by including a species' finite momentum, canonical flux tubes can become long and collimated even when the magnetic field is flared. Based on a generalized frozen-in-flux condition, these results suggest that transitions in the shape of astrophysical jets (i.e. flaring, bulging, and collimation) may reflect variation in the relative strength of the magnetic and fluid vorticity fluxes. Calculation of gauge-invariant relative canonical helicity shows that ion flow fields can unwind to compensate for the increasing magnetic twist and we hypothesize that conversion between various helicity components at the appropriate scales could convert helical magnetic twist into helical shear flows which could be sufficiently strong to stabilize against current driven instabilities.

To test this hypothesis, chapter 4 presents the design and initial results obtained during the commissioning phase and first operational phase of the MoCHI.LabJet experiment. During these first two phases, the initial set of diagnostics consisted of current and voltage probes, high-speed cameras, and a He-Ne interferometer for line integrated electron density measurements. During the commissioning phase, the experiment suffered from uncontrolled breakdown behind the gun electrodes and between the high voltage insulating gap separating the middle and outer electrodes as well as insufficient gas throughput to the outer electrode gas injection slits. To address these issues, a Teflon insulator was wrapped around the middle electrode re-entrant port and ten custom high-throughput gas valves were manufactured. Following these modifications, the first collimated jets were observed while operating the gun with both electrodes. These jets were short lived ($\sim 10\mu s$) with electron densities on the order of $10^{22} m^{-3}$, axial currents of $\sim$60kA, axial propagation velocities of $\sim$70 km/s, aspect ratios of $\sim 10:1$, and a stability consistent with the Kruskal-Shafranov condition. These first phase results replicated and confirmed the Caltech jet results [32, 34, 35, 73, 87]. Following detachment of these initial jets, we observed that the remaining energy in the capacitor banks appeared to drive a diffuse secondary plume bordered by a bright arch over the Teflon insulator. To improve

the performance of this secondary plume, the Teflon insulator was pushed further back into the re-entrant port and an alumina insert was mounted to the edge of the middle electrode. These modifications began the second operational phase and resulted in a drastic improvement. High speed video confirmed the formation of 1.1 m long jets with aspect ratios $\gtrsim 20{:}\,1$, electron densities of $\sim 10^{22}$ m^{-3}, core currents of 60 – 80 kA, skin currents of 100 – 120 kA with a pronounced helical structure. These long stable jets were formed after the short-lived initial Caltech-like jets. We call the initial Caltech-like jets "Stage I jets", the diffuse plume "Stage II jets or plumes" and the long stable jets "Stage III jets". These jets remained well collimated and stable for upwards of $\sim 40-60\,\mu s$, corresponding to > 40 radial Alfvén transit times, despite occupying classically unstable space. Temperature estimates assuming Spitzer resistivity using the current and voltage waveforms suggested that the temperature in the core was fairly constant at ~ 5 eV, while the skin temperature increased from ~ 5 eV to ~ 20 eV despite a decrease in current during this time.

To investigate the effect of flows and magnetic fields within the long stable jet, chapters 5 and 6 present the results from the third operational campaign that began with the completion and implementation of a custom optical fiber bundle for ion Doppler spectroscopy measurements and three custom magnetic probe arrays. Doppler broadening of the H_α emission line confirmed heating of the jet from $\sim$5 eV to $\sim$20 eV with a hollow temperature profile suggesting a cooler core plasma bordered by a hot plasma sheath in agreement with estimates from Spitzer resistivity. Observations of CII impurity lines revealed significant line integrated axial flow velocities on the order of $40-50$ km/s (near Alfvén velocities) and line integrated azimuthal velocities upwards of $\sim$20 km/s with clear indications of coherent azimuthal rotation consistent with the $\vec{E}\times\vec{B}$ drift direction. To infer local velocities, line integrated experimental measurements were compared to line integrated velocities through a double shell plasma model resulting in axial velocities on the order of $\sim$60 km/s, with a slight deceleration between $t = 49\mu s$ and $t = 59\mu s$ as the jet piles up against the end wall and the driving current decreases. In the azimuthal plane, model fits suggest local azimuthal velocities in the core of the jet on the order of $\sim$20 km/s with velocities in the skin of $\sim$8 km/s, both in the $+\hat{\phi}$ direction, consistent with the $\vec{E}\times\vec{B}$ direction of the LabJet gun. The resulting azimuthal flow shear was found to be $\sim$6 times larger than the theoretical threshold for wall-free stabilization [36]. To corroborate these shear flow estimates, a matrix Abel inversion was

performed on the same data set. The results of this inversion also suggested sufficiently strong azimuthal shear flows but with significantly enhanced velocities compared to the double shell model. Because many of the long jets appear to go unstable near the end of the discharge as the current ramps down to zero (when they should become more stable), this suggests that the $\vec{j} \times \vec{B}$ forces responsible for driving stabilizing shear flows may weaken more rapidly than the destabilizing effect of helical magnetic fields as the current decreases.

Magnetic field measurements suggest an additional mechanism for the enhanced stability observed in the long stable jets. While the magnetic field of the stage I jet is well described by a simple azimuthally symmetric screw pinch (which obeys the classical Kruskal-Shafranov instability limits), the magnetic field of the Stage III jet cannot be described as a screw pinch because there are strong, simultaneous positive and negative axial magnetic fields in the core of the jet which oscillate in time. Although high resolution measurements were limited to a single linear array, two and three-dimensional information was obtained by exploiting the natural oscillation of the jet in the transverse direction. The resulting analysis revealed the presence of left-handed positive and negative flux tubes with large antiparallel current densities and $\lambda a < -3$. These tubes appeared to wrap around each other in a right-handed helix and/or rotate in the $+\hat{\phi}$ direction around a common axis with a frequency of ~ 25 kHz. To test this hypothesis, experimental magnetic field traces were compared to hypothetical probe transits through a helically relaxed $m = 1$ non-axisymmetric double-helix Taylor state with an aspect ratio of 25:1 and a $\lambda a = -3.119$. The resulting field traces through the model are remarkable in their similarity to the experimental data, suggesting that the magnetic field structure of the long, stable, collimated jet is in qualitative agreement with a helically relaxed non-axisymmetric double-helix Taylor state despite the absence of a nearby flux conserving wall. The magnetic structure of the MoCHI.LabJet is similar to the double-helix state produced in the SSX experiment with a close-fitting metal flux conserver at aspect-ratios of 3:1 and 13.4:1 [17, 99].

The measurements presented in this thesis contribute towards a comprehensive understanding of the stable jets formed in the MoCHI.LabJet experiment (Figure 7.1). The observations suggest that the core of the jet consists of a double helix magnetic field qualitatively similar to a Taylor double helix which rotates in the $+\hat{\phi}$ direction like an Archimedes screw with a negative helicity content in agreement with the $\vec{E} \times \vec{B}$ of the gun. This structure appears to be enclosed by a hot sheath plasma which also rotates in the same direction but with a reduced

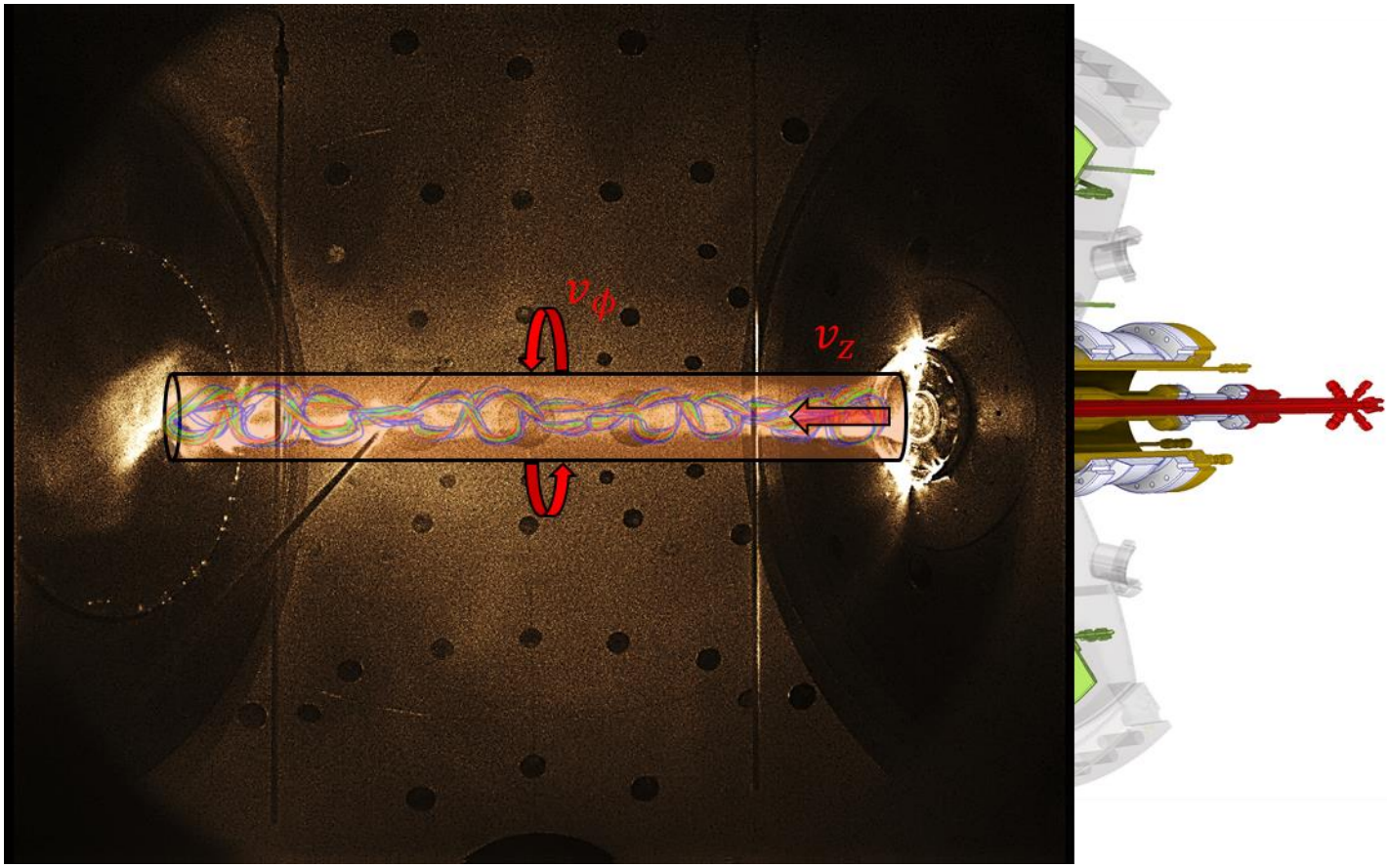


Figure 7.1. Comprehensive picture of the long, stable, collimated jets formed in the MoCHI.LabJet experiment based on experimental observations presented in this thesis. The core of the jet consists of an isolated, high-β, double-helix relaxed state magnetic field with negative helicity content which rotates in the $+\hat{\phi}$ direction like an Archimedes screw. This structure is bordered by a hotter sheath plasma (orange) that rotates in the azimuthal direction with a lower velocity than the core. Strong positive axial flows are also observed and are presumed to flow within both volumes. The helical shear flow is sufficient to stabilize the current-carrying magnetized jet to kink instabilities.

velocity. Large positive axial flow velocities on the order of the Alfvén velocity are also observed within the jet. Such strong flows and a plasma β on the order of unity are inconsistent with a helically relaxed, static, zero β Taylor state; however, previous experiments have demonstrated good quantitative agreement for plasmas with a $\beta \approx 0.4$ and flows on the order of the ion sound speed within a cylindrical flux conserver with an aspect ratio of 13.4 [99]. The unbounded nature of the jets in the MoCHI.LabJet experiment however, suggest a unique, and heretofore unobserved plasma state that may be better described as a generalized, non-Taylor, driven

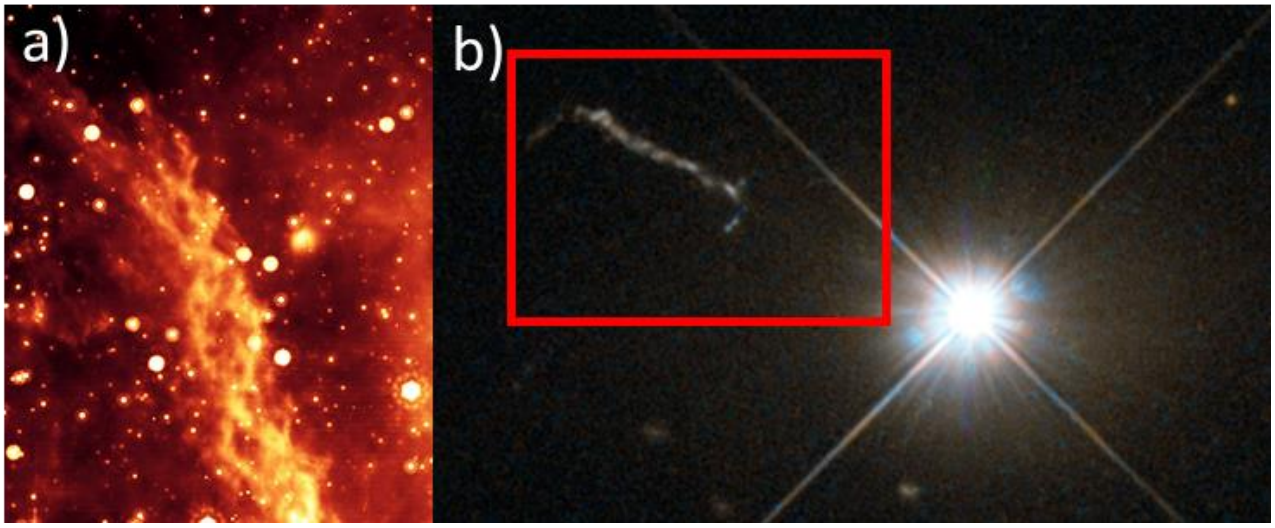


Figure 7.2. Astrophysical jets with a double helix structure. a) The Double Helix Nebula near the center of our own Milky Way Galaxy. b) Quasar 3C273 with a ~200,000 light year long helical jet boxed in red. Image credits: Nasa/JPL – Caltech/M. Morris (UCLA), ESA/Hubble.

equilibrium. Such a state may be particularly useful and attractive for future magnetically confined fusion concepts based on the high density, high β, high-field stability without close-fitting walls. The concept is also attractive for use as a plasma thruster with a high specific impulse and low (if not zero) divergence angle. Furthermore, the observation of a double helix within our laboratory jet experiment is mirrored in astrophysical observations of several jets thought to be associated with galactic supermassive black holes. One such jet, known as the double helix nebula (Figure 7.2a) is located near the center of our own galaxy and was speculated to arise from torsional Alfven waves [100]. A second jet, quasar 3C273 (Figure 7.2b), was also observed to have a double helix structure which was speculated to arise due to a Kelvin-Helmholtz instability [101]. Our results present an alternative explanation and suggest that, compared to previous planar plasma gun experiments, our unique triple electrode setup with azimuthally symmetric gas slits more accurately reproduces the essential astrophysical boundary conditions required to produce incredibly long, stable, collimated jets in an unbounded volume.

Chapter 8
SUGGESTIONS FOR FUTURE WORK

There are numerous avenues for future investigation on MoCHI experiment. First and foremost, to live up to its acronym (Measurement of Canonical Helicity Injection), volumetric measurements of the velocity and magnetic field evolution is required with sufficient fidelity to approach the ion inertial length. This could be accomplished by improving the shot-to-shot reproducibility and increasing the number of diagnostic channels. Recently five additional FPGA digitizers have become available which doubles the number of magnetic probe channels that can be measured in a single shot. Furthermore, the development of additional optical fiber bundles can be accomplished in cost effective manner following appendix B. To improve shot to shot reproducibility, it is critical to improve the insulation between the middle and outer electrodes. In this thesis, it was discovered that an alumina insert was required to form the long, collimated, stable jets. Arcing over this insert however, introduced significant shot to shot variability and eventually resulted in its destruction (possibly due to heating of the Teflon behind/underneath it). Once consistent volumetric measurements are obtained, the resulting canonical flux tubes and canonical helicity content within the jet can be reconstructed following reference [63]. Tracking the evolution of the various helicity components could be used verify the theory of canonical helicity transport [50] by showing that species helicities are coupled while the total canonical helicity is conserved.

Further operation could continue a sweep of the extensive parameter space available on this machine. Experimental control knobs that can be varied include the voltage of the inner and middle electrode (swept briefly during operations 2 and 3 but not extensively), firing times of the gun ignitrons, bias magnetic field strength, gas injection timings, throughput, and gas type. Furthermore, it may be worthwhile to investigate the end wall boundary conditions of the floating spiderleg gun electrode with high voltage probes. Currently, the jet length appears to be limited only by the floating end wall. Replacing the spiderleg gun with a longer vacuum chamber would be worthwhile to investigate whether MoCHI can generate jets significantly longer than 1 m. To increase the duration of the pulse, the LabJet power supplies can be easily modified to form a pulse forming network. With a flat top current, it should also be easier to measure the energy confinement time.

Modifying the orientation of various diagnostics could also prove to be extremely useful towards resolving unanswered questions. For example, mounting optical fiber bundles on opposite sides of the vacuum chamber could help pin down a reference zero velocity and verify that the observed negative skew in H_α emission is in fact due to radial outflows. Provided sufficient lines of sight, flow velocities within the chamber could even be reconstructed using vector tomographic reconstruction without any assumptions of symmetry. An axially aligned magnetic probe could measure the pitch of the double helix and allow us to differentiate between rotation and axial translation while providing an additional point of comparison between a theoretical $m = 1$ Taylor state [99].

Finally, from a computational perspective, development of a generalized multifluid equilibria solver that does not require assumptions of symmetry would represent a tremendous advancement. The results from such a solver could be used to compare to experimental results obtained on the MoCHI experiment. Furthermore, a full dynamical simulation of the LabJet experiment would be also be valuable and discussion with several parties interested in running such a simulation is ongoing.

BIBLIOGRAPHY

1. J. A. Bittencourt. *Fundamentals of Plasma Physics*. New York: Springer-Verlag, 2004. DOI: 10.1007/978-1-4757-4030-1.
2. D. A. Gurnett and A. Bhattacharjee. *Introduction to Plasma Physics*. Cambridge, UK: Cambridge University Press, 2005. DOI: 10.1017/cbo9780511809125.
3. P. M. Bellan. *Fundamentals of Plasma Physics*. Cambridge, UK: Cambridge University Press, 2006. ISBN: 978-0-521-52800-9. DOI: 10.1017/cbo9780511807183.
4. Woltjer, L. 1958. Proc. Nat. Acad. Sci. U.S.A. 44, 489-491.
5. P. M. Bellan. *Spheromaks*. Imperial College Press, 2000. ISBN: 978-1-86094-141-2.
6. Finn, J. M., & Antonsen, T. M. 1985, CoPPC, 9, 111.
7. Taylor, J.B. 1974, PhRvL, 33, 1139
8. Lundquist, S., Magnetohydrostatic fields, *Ark. Fys., 2, 361, 1950.*
9. Chandrasekhar, S., On force-free magnetic fields, *Proc. Natl. Acad. Sci. U.S.A.*, 42, 1, 1956.
10. Chandrasekhar, S., and P. C. Kendall, On force-free magnetic fields, *Astrophys. J.*, 125, 457, 1957
11. Rosenbluth, M. N., and M. N. Bussac, MHD stability of spheromak, *Nucl. Fusion, 19,489*, 1979
12. P. M. Morse and H. Feshback, *Methods in Theoretical Physics* (McGraw-Hill, New York, 1953), Vol. 2, 1776.
13. J. M. Finn, W. Manheimer, and E. Ott,. 1981. Phys. Fluids 24, 1336.
14. A. Bondeson, G. Marklin, Z. G. An, et al. 1981. Phys. Fluids 24, 1682.
15. J. B. Taylor, Rev. Mod. Phys. 58, 741 (1986).
16. M. J. Schaffer, Phys. Fluids 30, 160 (1987).
17. C. D. Cothran, M. R. Brown, T. Gray. et al. 2010. PhPl 17, 055705.
18. Turner, L.1986, IEEE Trans Plasma Sci. 14 849
19. Den Hartog D, Almagri, A., Chapman, J., Ji, H., Prager, S., & Sarff, J., 1995 PhPl, 2, 2281
20. Ono, Y., Yamada, M., Akao, T., Tajima, T., & Matsumoto, R. 1996, PhRvL, 76, 3328
21. DeYoung, D. 1991, Sci, 252, 5004.
22. R. Sahai, M. Morris, G. Knapp, K. Young, & C. Barnbaum. 2003, Natur, 426, 6964.

23. C. Burrows, K. Stapelfeldt, A. Watson, et al. 1996, ApJ, 473, 1.
24. M. Nakamura, Y. Uchida, & S. Hirose. 2001, NewA, 6, 2.
25. Icke, V. Mellema, G. Balick, B. Eulerink, F., & Frank, A.1992, Natur, 355, 6360.
26. B. Remington, R. Drake and D. Ryutov, Rev. Mod. Phys., 78, 755-807, (2006).
27. Norman, M., Smarr, L., Winkler, K. & Smith, M. 1982, A&A, 113, 2.
28. Frank, A., Ryu, D., Jones, T. W., & Noriega-Crespo, A. 1998, ApJ, 494, 79.
29. Broderick, A., & Loeb A. 2009, ApJ, 703, 104.
30. Donati,J., Howarth, I., Jardine, M.,Petit, P., et al. 2006, MNRAS, 370, 629.
31. D. Lynden-Bell, 2003. MNRAS., 341, 4.
32. Bellan, P., You, S., Hsu, S. C. 2003. Astrophys. & Space Sci., 298, 203-209.
33. Bellan, P. 2003, PhPl, 10, 5.
34. You, S., Yun, G., & Bellan, P. 2005, PhRvL, 95, 4.
35. Yun, G., You, S., & Bellan, P. 2007, NucFu, 47, 3.
36. Shumlak, U., & Hartman, C. 1995, PhRvL, 75, 18.
37. Shumlak, U. Nelson, B., Golingo, R., et al. 2003, PhPl, 10, 5.
38. Burlaga, L. F., E. Sittler, F. Mariani, and R. Shwenn, Magnetic loop behind an interplanetary shock: Voyager, Helios and IMP-8 observations, *J. Geophys. Res., 86, 6673, 1981*
39. Bothmer, V., *and* R. Shwenn , The structure and origin of magnetic clouds in the solar wind, *Ann. Geophys., 16, 1–24. 1998*
40. Farrugia, C. J., V. A. Osherovich, and L. F. Burlaga (1995), Magnetic flux rope versus the Spheromak as models for interplanetary magnetic clouds, J. Geophys. Res., 100(A7), 12293–12306.
41. Osherovich, Vladimir ; Fainberg, Joseph ; Webb, Alla. 2013. Solar Physics, 284(1), pp.261-274
42. Richardson, I. G., *and* H. V. Cane , The fraction of interplanetary coronal mass ejections that are magnetic clouds: Evidence for a solar cycle variation, *Geophys. Res. Lett., 31, L18804, 2004*
43. E. Priest. *Magnetohydrodynamics of the Sun* (Cambridge University Press, Cambridge, 2014)
44. C. J. Schrijver, M. L. DeRosa, A. M. Title, and T. R. Metcalf, ApJ 628, 501 (2005)

45. Ono, Y., Yamada, M., Akao, T., Tajima, T., & Matsumoto, R. 1996, PhRvL, 76, 3328
46. Kawamori, E. et al. 2005, NucFu, 45, 843.
47. Steinhauer, L., & Ishida, A. 1998, PhPl, 5, 2609.
48. You, S. 2016, PhPl, 23, 072108.
49. Brown, M. 1997, PlPh, 57, 1.
50. You. S. 2012, PhPl, 19, 092107.
51. Steinhauer, L., & Ishida, A. 1997, PhRvL, 79, 3423.
52. Ji, H. 1999, in Magnetic Helicity in Space and Laboratory Plasmas, ed M. Brown et al. (Washington, DC: American Geophysical Union), 167.
53. Moffat, H. 1969, JFM, 35, 117.
54. Steinhauer, L., Yamada, H., & Ishida, A. 2001, PhPl, 8, 4053.
55. Turner, L.1986, IEEE Trans Plasma Sci. 14 849
56. Avinash, K. 1992, PhFlB, 4, 3856.
57. Oliveira, S. R., & Tajima, T. 1995, PhRvE, 52, 4287.
58. Lavine, E.S., You, S. 2017, ApJ, 835, 1
59. Jarboe, T. R., Hamp, W. T., Marklin, G. J., et al. 2006, PhRvL, 97, 115003.
60. You, S. 2014, PPCF, 56, 064007.
61. Fiksel, G., Almagri, A. F., Chapman, B. E., et al. 2009, PhRvL, 103, 145002.
62. Ray, T. P., & Mundt, R. 1993, in Astrophysical Jets, ed D. Burgarella et al. (Caimbridge: University Press), 149.
63. Jens von der Linden. 2017 Investigating the Dynamics of Canonical Flux Tubes (Doctoral dissertation). University of Washington.
64. Bouquet, S., et al. 2010, HEDP, 6, 368
65. Boehly, T., et al. 1997, OptCo, 133, 1, 495
66. Foster, J. et al. 2005, ApJ, 634, 1, L11
67. Hogan, W., et al. 2001, NucFu, 41, 5, 567
68. Besnard, D. 2007, EPJD, 44, 2, 207
69. Ampleford, D., et al. 2008, PhRvL, 100, 3, 035001
70. T. Byvank, J. Chang, W. M. Potter, et. al 2016. IEEE Trans. Pl. Sci., 44, 4
71. Ryutov, D. D., et al. 2000, ApJ Sup., 127:465-468
72. Lebedev, S. V., et al. 2002, ApJ, 564, 113

73. Hsu, S., & Bellan, P. 2003, PhRvL, 90, 21, 215002
74. G. Bateman. *MHD Instabilities*. Cambridge: The MIT Press, 1978.
75. Hartigan P. , et. al ApJ, 661:910-918, 2007.
76. Ciardi, A.. Lect. Notes Phys. 793, 31–50 (2010)
77. S. You et all. ApJ. 2018
78. Evan Carroll "Driving flows in laboratory astrophysical plasma jets: The Mochi.Labjet experiment". MA thesis. University of Washington 2016.
79. Hsu, S. C., & Tang, X. Z. 2008, BAPS, 53, 271
80. Auna L Moser. "Dynamics of magnetically driven plasma jets: An instability of an instability, gas cloud impacts, shocks, and other deformations". PhD thesis. California Institute of Technology, 2012.
81. Chaplin, V., & Bellan, P. M. 2013, ITPS, 41, 4, 975
82. von der Linden, J. 2017, MoCHIFPGAcontrol: High throughput FPGA code for NI-5752 digitizers, Zenodo, doi:10.5281/zenodo.495787
83. von der Linden, J., Card, A., & You, S. 2017, MoCHIControl: LabVIEW experiment control with MDSplus data storage, Zenodo, doi/10.5281/zenodo.495643
84. Stillerman, J. A., et al. 1997, RScI, 68, 1, 939
85. Manduchi, G., et al. 2013, Fusion Engineering & Design, 88, 6, 1196
86. Card, A. H., 2017, Implementation of an unequal path length heterodyne interferometer on the MoCHI LabJet experiment (Seattle, WA, University of Washington)
87. Hsu, S. C., Bellan, P. 2005, PhPl, 12 032103.
88. Moser, A. L., & Bellan, P. M. 2012, Natur, 482, 379
89. von der Linden, J., & You, S. 2017, PhPl, 24, 052105
90. I.H. Hutchinson. Principles of plasma diagnostics. Cambridge university press, 2005.
91. W. C. Martin, J. R. Fuhr, D. E. Kelleher, A. Musgrove, L. Podobedova, J. Reader, E. B. Saloman, C. J. Sansonetti, W. L. Wiese, P. J. Mohr, and K. Olsen. NIST atomic spectra database (version 2.0), [online]. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 2002. Available:
92. Yun, G., You, S., & Bellan, P. 2007, NucFu, 47, 3, 181
93. Bell, R. 1995, RScI,66, 558
94. Bell, R. 1997, RScI,68, 2

95. A. B. Filuk and J. E. Bailey. Analyzing noisy spectral line shapes. Review of Scientific Instruments, 63(10), 4783-4785 1992
96. Wanex, L. F. et al. 2005, PhPl 12, 042101
97. J. D. Jackson, Classical Electrodynamics, John Wiley & Sons, New York, NY. 1975, p298
98. Hu, Q., and B. U. O¨ . Sonnerup, Reconstruction of magnetic flux ropes in the solar wind, Geophys. Res. Lett., 28, 467 – 470, 2001.
99. T. Gray, M. R. Bwon, & D. Dandurand. 2013, PhRvL, 110, 085002.
100. Morris, M. Uchida, K. Do, T. 2006. Nature, 440, 7089.
101. Lobanov A. P. Zensus. J. A. 2001. Science, 295, 5540. DOI: 10.1126/science.1063239

APPENDIX A

FAST IONIZATION GAUGE FOR NEUTRAL GAS DENSITY MEASUREMENTS

This appendix describes the development of a fast ionization gauge (FIG) designed to ensure adequate and symmetric gas injection on the MoCHI.LabJet gun. To track the evolution of a gas puff, the probe has a temporal resolution $< 2\mu s$ and can measure pressure changes from 10^{-7} to 10^{-3} Torr.

A.1. *Operation*

The FIG is a hot cathode gauge designed to perform rapid measurements of the neutral gas density. It operates on the basis of ionizing gas molecules in the gauge and collecting the resulting ion current in a collector circuit where the magnitude of the current can be measured.

To ionize gas molecules, a negatively biased electrode (tungsten filament) is ohmically heated to emit a stream of electrons. These electrons ionize neutral gas molecules before colliding with an open anode cage structure. Positive ions are attracted to the most negatively biased element in the detector head, the ion collector plate. The resulting ion current is proportional to the gas density and is measured using a differentiating op-amp.

A.2. *Principle*

The number of ions formed near the detector head, and thus the ion current that is measured by the op-amp circuit is a function of numerous variables:

- The number of molecules per unit volume
- The ionization cross section for the particular gas at a specified electron energy
- The electron emission/arrival rate (i.e. emission current)
- The path length of the electrons

Defining σ_i to be the ionization cross section of a gas molecule, L the length of the ionizing space, and A to be the cross-sectional area of the electron beam, the number of molecules included in this volume is given by nLA where n is the molecular density (related to pressure by $n = P/k_BT$). The total ionization cross section for molecules in this volume is $A_\sigma = nLA\sigma_i$ and the fraction of electrons that participate in ionizing collisions is

$$\frac{A_\sigma}{A} = nL\sigma_i = L\sigma_i \frac{P}{k_BT}. \tag{A.1}$$

If we define N to be the number of electrons entering the anode cage per unit time, the number of ionizations per unit time is $NL\sigma_i\, P/k_BT$. Assuming all ions are collected by the collector pin, the resulting ion current is

$$I_C = NL\sigma_i \frac{P}{k_BT} e \tag{A.2}$$

where e is the electron charge. Substituting the expression for the electron emission current ($I_e = Ne$) one obtains

$$I_c = \left[L\sigma \frac{1}{k_BT}\right] I_e P \tag{A.3}$$

where the factor in brackets is a function of gas type, geometry, and absolute temperature and is generally defined as the sensitivity factor S. This reduces the equation to the standard ionization gauge equation

$$P = \frac{I_c}{S\, I_e}. \tag{A.4}$$

Oftentimes, it is useful to break up the sensitivity factor into two components, the sensitivity factor for air and a relative sensitivity factor for the gas being measured ($S = S_{air}R_g$). It is important to note however, that the only reliable way of obtaining the sensitivity factor is through a direct and careful calibration.

A.3. *Electrical Design*

The electrical design of this device can be broken into two separate circuits, the filament (cathode)-anode grid circuit and the ion collector circuit (Figure A.1). The first of these circuits is quite simple, consisting of a 47 V battery to negatively bias the tungsten filament and an external power supply. The external power supply generates a current through the filament which then heats ohmically. When the temperature of the filament reaches what is required for thermionic emission, electrons are boiled off the surface and accelerate towards the grounded anode grid. Two BNC ports along the circuit allow for measurement of the bias current, and more importantly the emission current.

The second circuit resembles that of a differentiator op-amp. This design is advantageous because it allows for an output signal that directly relates to the magnitude of the ion current captured by the collector pin. A simple circuit analysis confirms this, where an ideal differentiator has an output defined by

$$V_{out} = -R_1 C_1 \frac{dV_{in}}{dt} \tag{A.5}$$

where R_1 and C_1 correspond to the feedback resistor and lead capacitor. Because the current is defined as the time rate of change of electric charges (dQ/dt), and the voltage across the input capacitor is defined as $V_{in} = Q_1/C_1$, it can be seen that

$$\frac{dV_{in}}{dt} = \frac{1}{C_1}\frac{dQ}{dt} = \frac{1}{C_1} I_c \, . \tag{A.6}$$

Thus, for the ideal differentiator op-amp

$$V_{out} = -R_1 I_c. \tag{A.7}$$

Of course, for the non-ideal case, a slight current drawn by the op-amp will lead to a small bias voltage included in the output voltage.

For this design the AD711 operational amplifier was selected. The AD711 has a 25 mA maximum output current which means it should be capable of driving up to 1.25 V

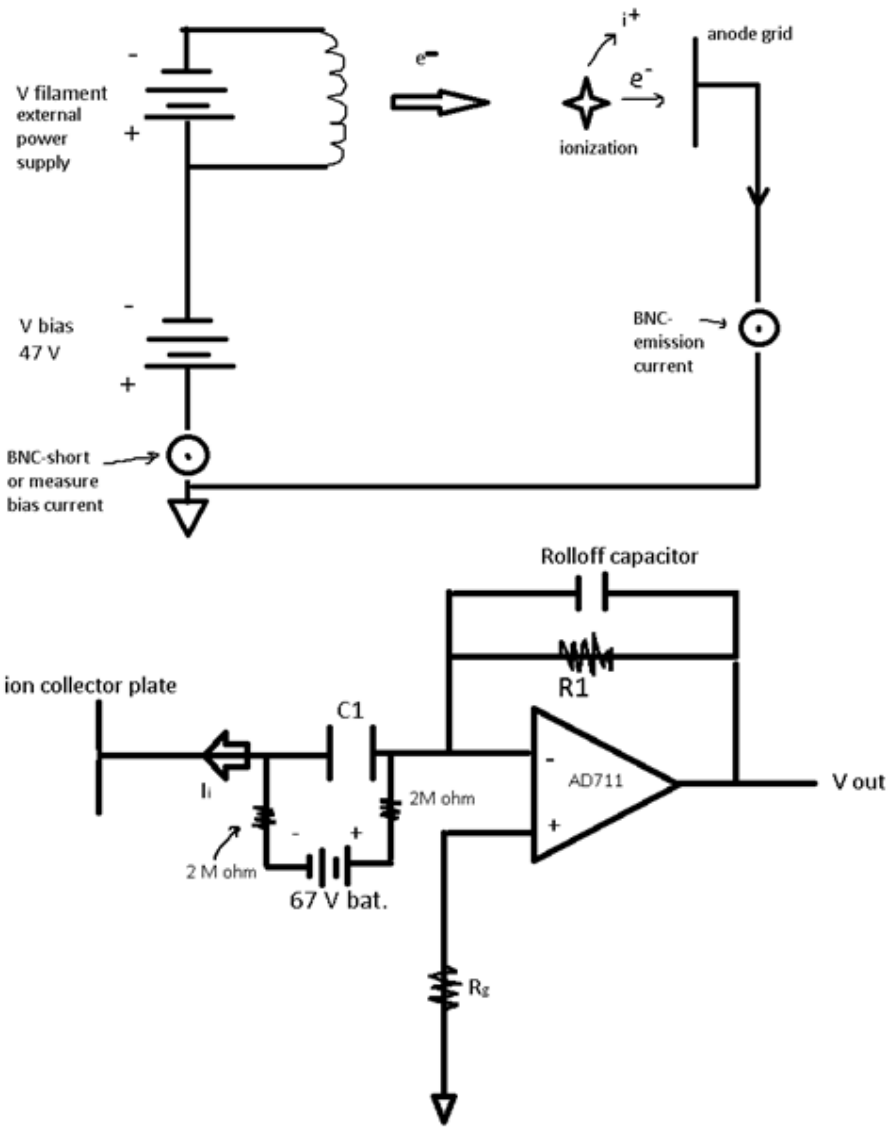


Figure A.1. Fast ionization gauge cathode-anode circuit (above). Simplified differentiator op-amp circuit for ion current measurements (below).

into a 50 Ω load. This op-amp was selected because it has an extremely low input offset current (20 pA). Additionally, the input leads are made as short as possible so as to reduce the rise time to $<2\mu s$.

A.4. *Vacuum Design*

When considering the vacuum components to use for this device, both the mobility of the detector head (for measurements across the planar electrodes) as well as the ability to maintain a high vacuum during these movements needed to be considered. A 3/4" outer diameter tube with a 0.62" inner diameter and a VCO (O-ring) seal welded to one end was used to mount the detector head. A ceramaseal 8-pin instrument feed through was secured to this VCO connector via a nut fastener. Three hairpin bends of tungsten wire (approximately 2 cm in length) were spot welded to these pins, along with the molybdenum ion collector plate and nickel anode cage. The gauge was mounted to the chamber using a 2.75" quick disconnect conflat. To aid in linear and rotational mobility, a flange-mount linear sleeve bearing was secured to the quick disconnect. The vacuum tube terminates in the FIG box where the bias batters and op-amp power supply are stored. The amplifier circuit is mounted on a miniature PCB board located inside the bacuume tube just inches away from the detector head itself. The short distance between the gauge and amplifier reduce capacitance in the leads and improves amplifier rise time. A solid works model and picture of this design are presented in figure A.2.

A.5. *Calibration*

Calibration of the FIG was performed in-situe in the MoCHI vacuum chamber using a standard Bayard-Alpert gauge as a reference. For each calibration test, the gate valve of the cryopump was closed and the chamber was allowed to rise from a base pressure of $\approx 5 \times 10^{-7}$ Torr to $10^{-3}\,Torr$ (the operational ceiling for the probe before the filaments would oxidize and burn up). The output voltage, V_{out}, and emission current I_e were measured throughout the pressure rise, as was the output of the Bayard-Alpert Gauge. The resulting calibration curves of V_{out}/I_e as a function of chamber pressure (Figure A.3) revealed a non-linear relationship not predicted by equations A.4 and A.7. The curves were fit with a power law curve using a least squares method and method that minimized the total percent error. The least squares fit did not weight low pressure values equally so the minimized percent error curve parameters for A and B were utilized while the correct

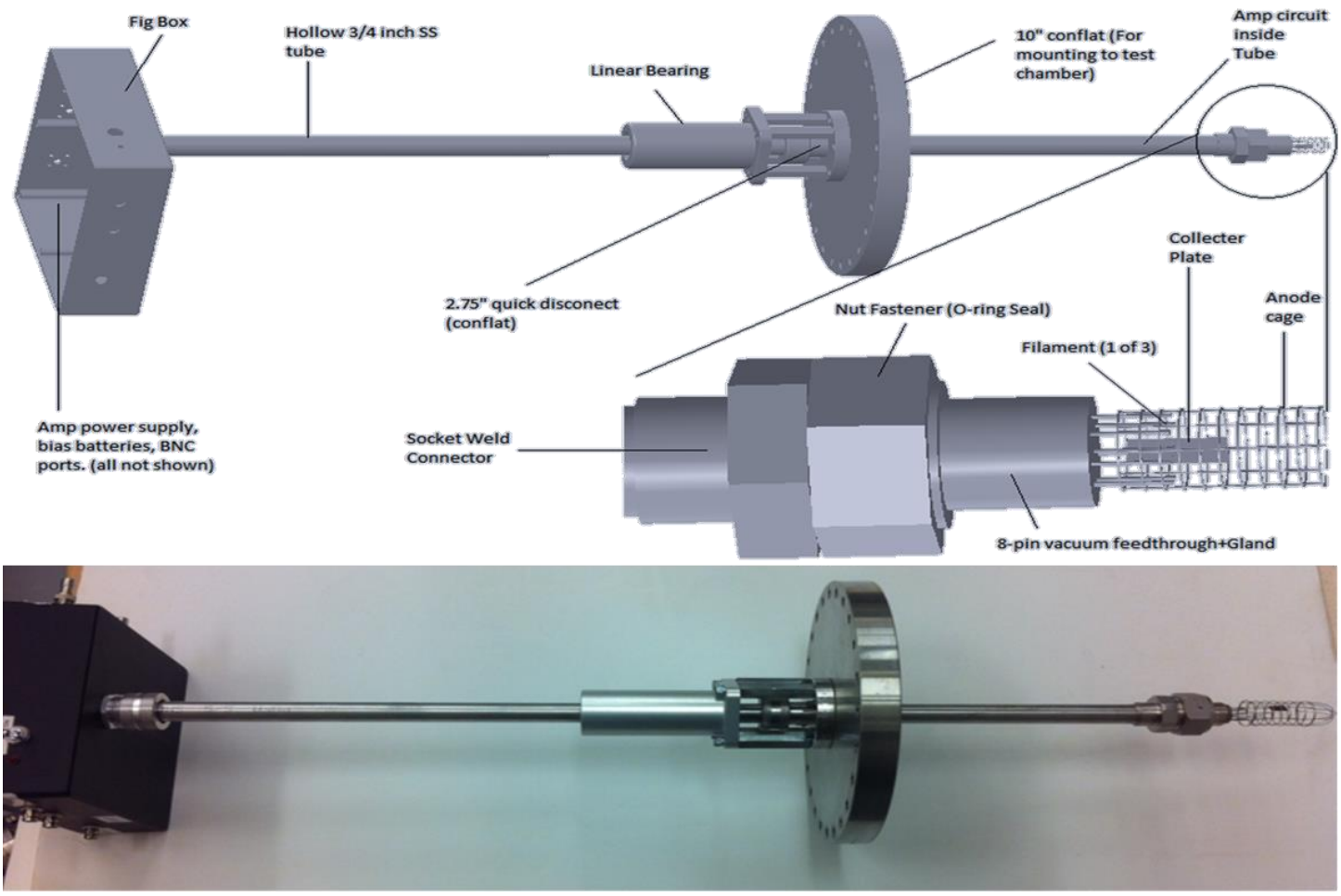


Figure A.2. Solid works diagram of the fast ionization gauge vacuum design (above). Corresponding picture of the completed probe (below).

calibration parameter for the offset C was calculated from shot to shot based on the initial base pressure P_0 and $(V_{out}/I_e)_0$ value as follows

$$C = \left(\frac{V_{out}}{I_e}\right)_0 - A(P_0^B). \tag{A.8}$$

The resulting pressure could then be reconstructed as

$$P = \left(\frac{1}{A}\right)\left(\frac{V_{out}}{I_e} - C\right)^{\frac{1}{B}}. \tag{A.9}$$

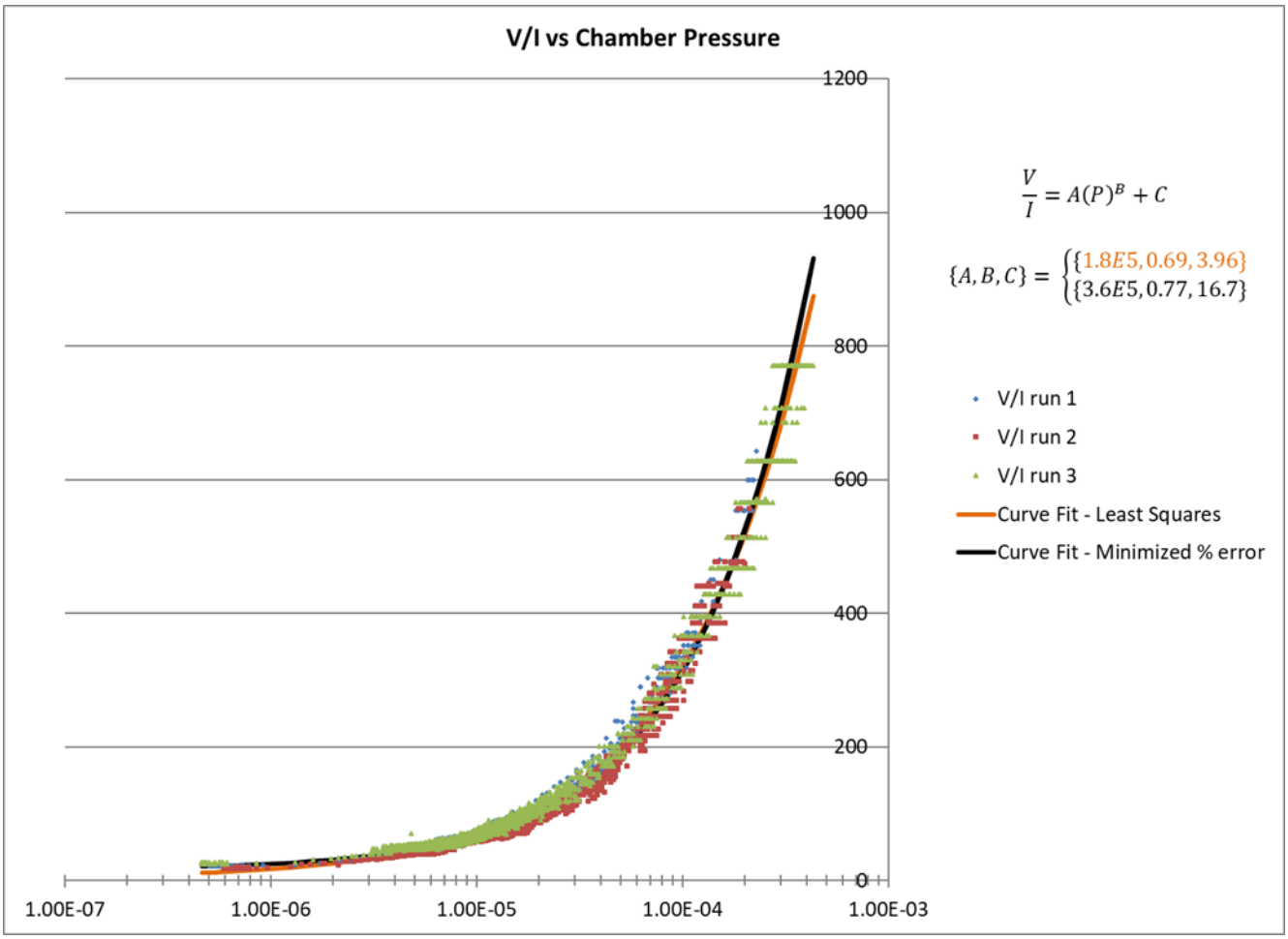


Figure A.3. Calibration curve of V_{out}/I_e as a function of chamber pressure for three calibration tests. Curves were not linear as expected, and were best fit using a power law curve that minimized the total percent error such that the low-pressure values were weighted equally.

Figure A.4 shows the results of the FIG reconstructed pressure for each of the three calibration runs. The results demonstrate that the FIG reconstructed pressures are quite accurate from $\sim 5 \times 10^{-7}$ to 3×10^{-4} Torr.

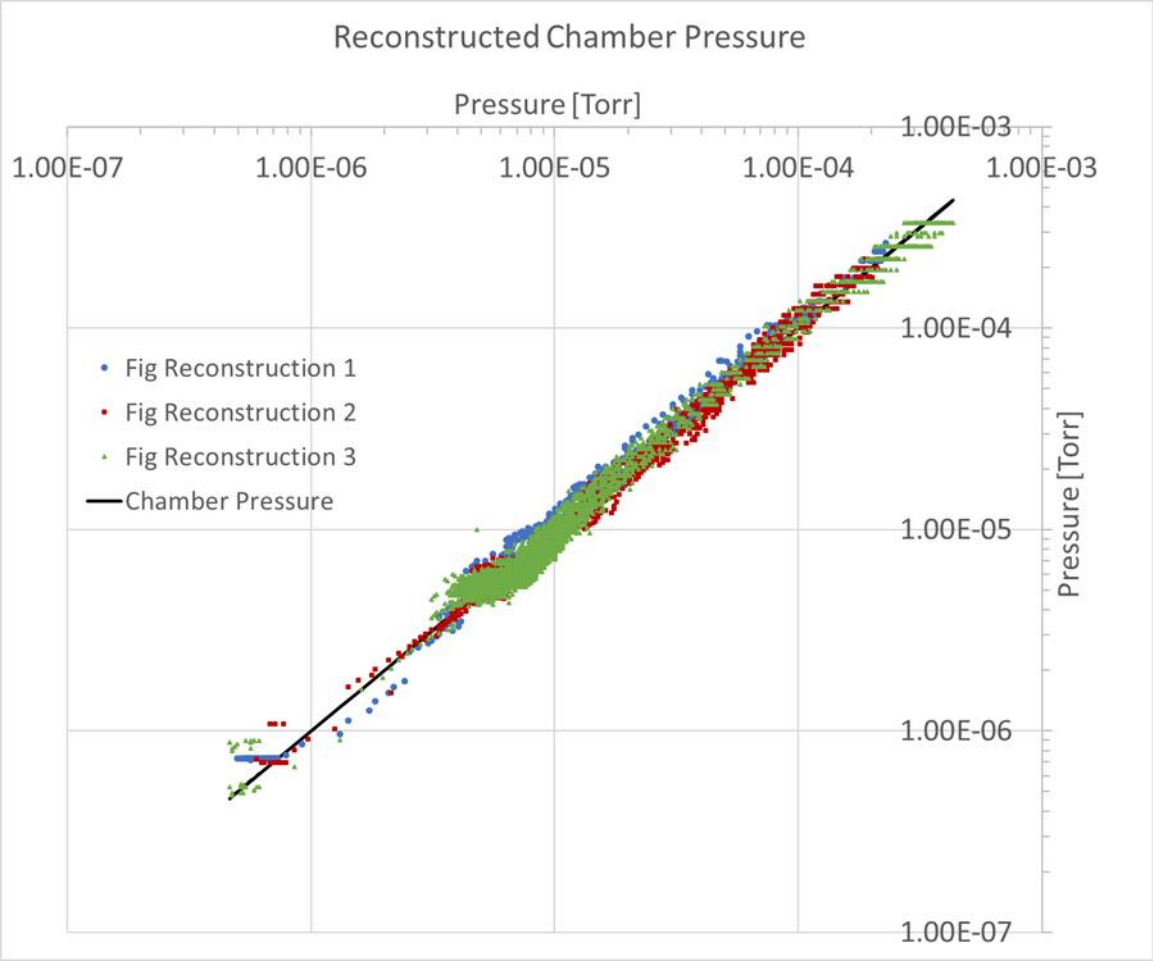


Figure A.4. Reconstructed FIG pressures for each of the three calibration runs. The Reconstructed pressures are in good agreement with the actual chamber pressure over a range of $\sim 5 \times 10^{-7}$ - 3×10^{-4} Torr.

A.6. *Characterization of fast gas valves*

Following calibration, the fast ionization gauge was used to characterize fast gas valves including commercially available Parker series 9 valves (Figure A.4) as well as the ten custom fast gas valves built in house based on valves inherited from Los Alamos National Laboratory (Referred to as Bellan Valves) (Figure A.5). The FIG was also used to deduce

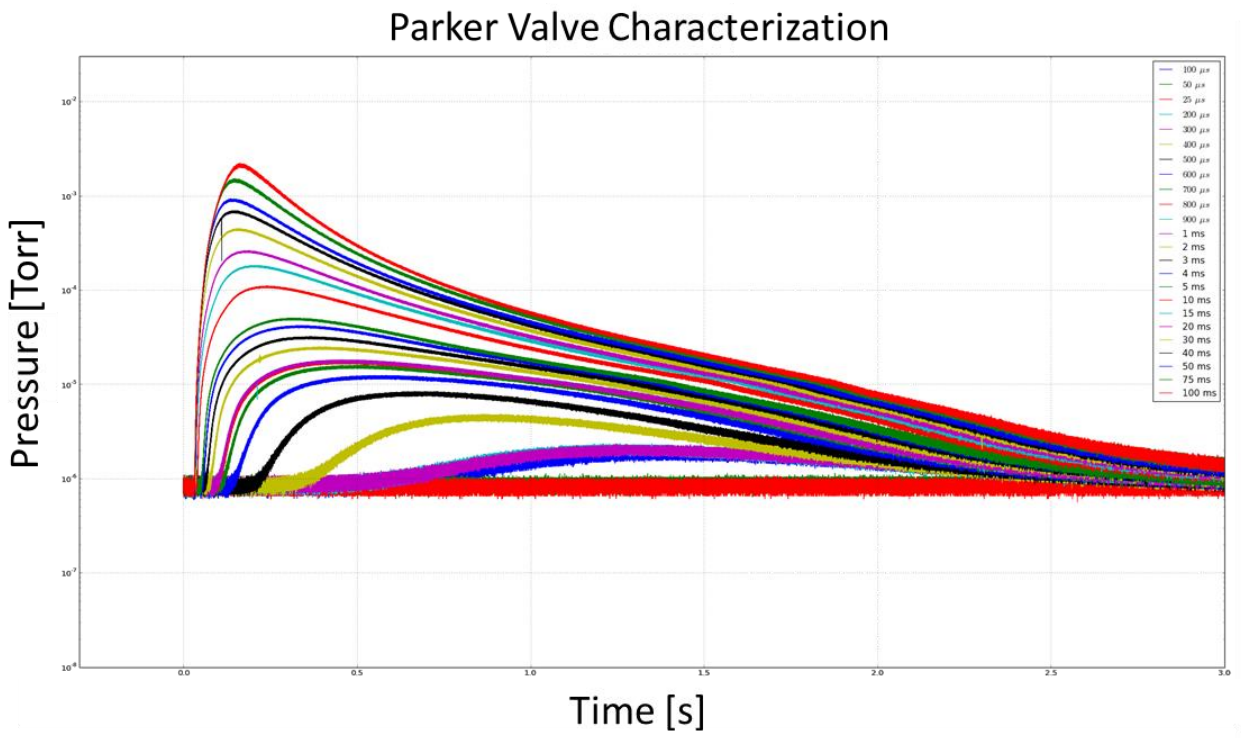


Figure A.5. Characterization of the Parker Series 9 valve driven by a custom fast valve drive unit (FVDU). Different pulse lengths resulted in differing throughputs and opening times, however all proved to be insufficient.

the appropriate gas injection firing times during the operational phase of the MoCHI.LabJet. To examine the time evolution of neutral gas density, figure A.6 shows the results of a FIG sweep across the electrodes while firing a single valve attached to the middle electrode plenum and a single valve connected to all outer-inner gas injection ports. During the operational phase, the FIG was also used to deduce the total particle inventory injected into the chamber, whic was determined to be $(7.8 \pm 0.6) \times 10^{23}$ particles for hydrogen with a backline pressure of 100 psi. This corresponds to a total neutral density of $(6.8 \pm 0.6) \times 10^{22} m^{-3}$ over the whole chamber [86].

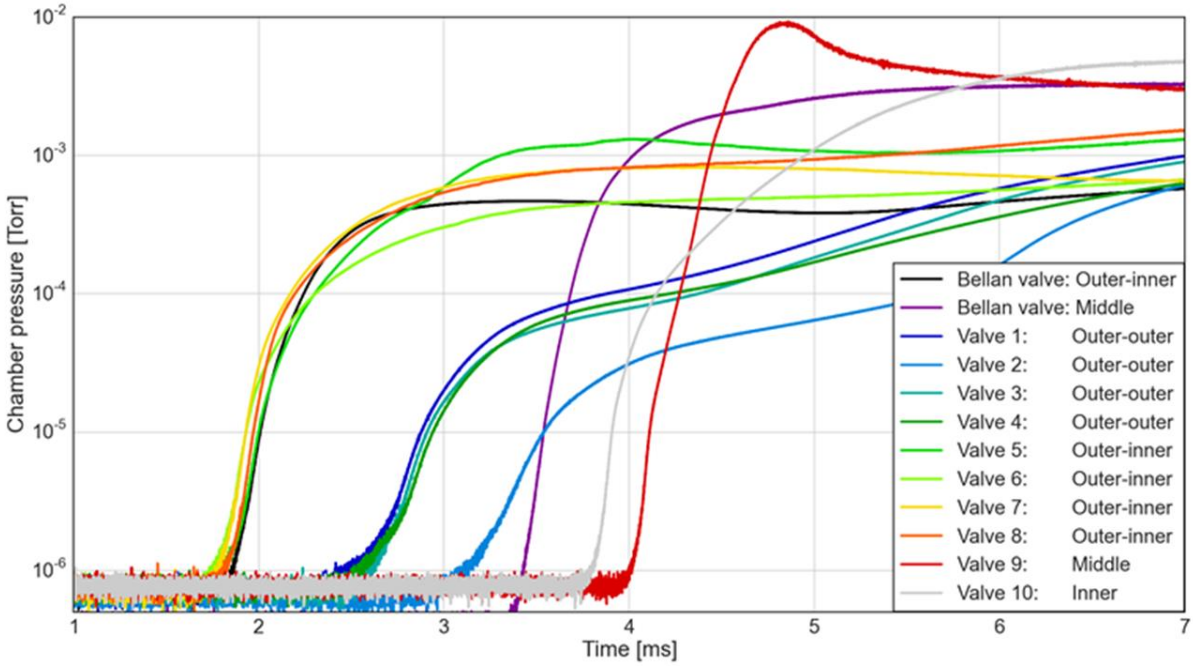


Figure A.6. Characterization of the 10 custom fast gas valves at their respective operational positions compared to the Bellan valves after which they were designed. For this test the FIG detector head was approximately 15 cm in front of the center of the LabJet gun.

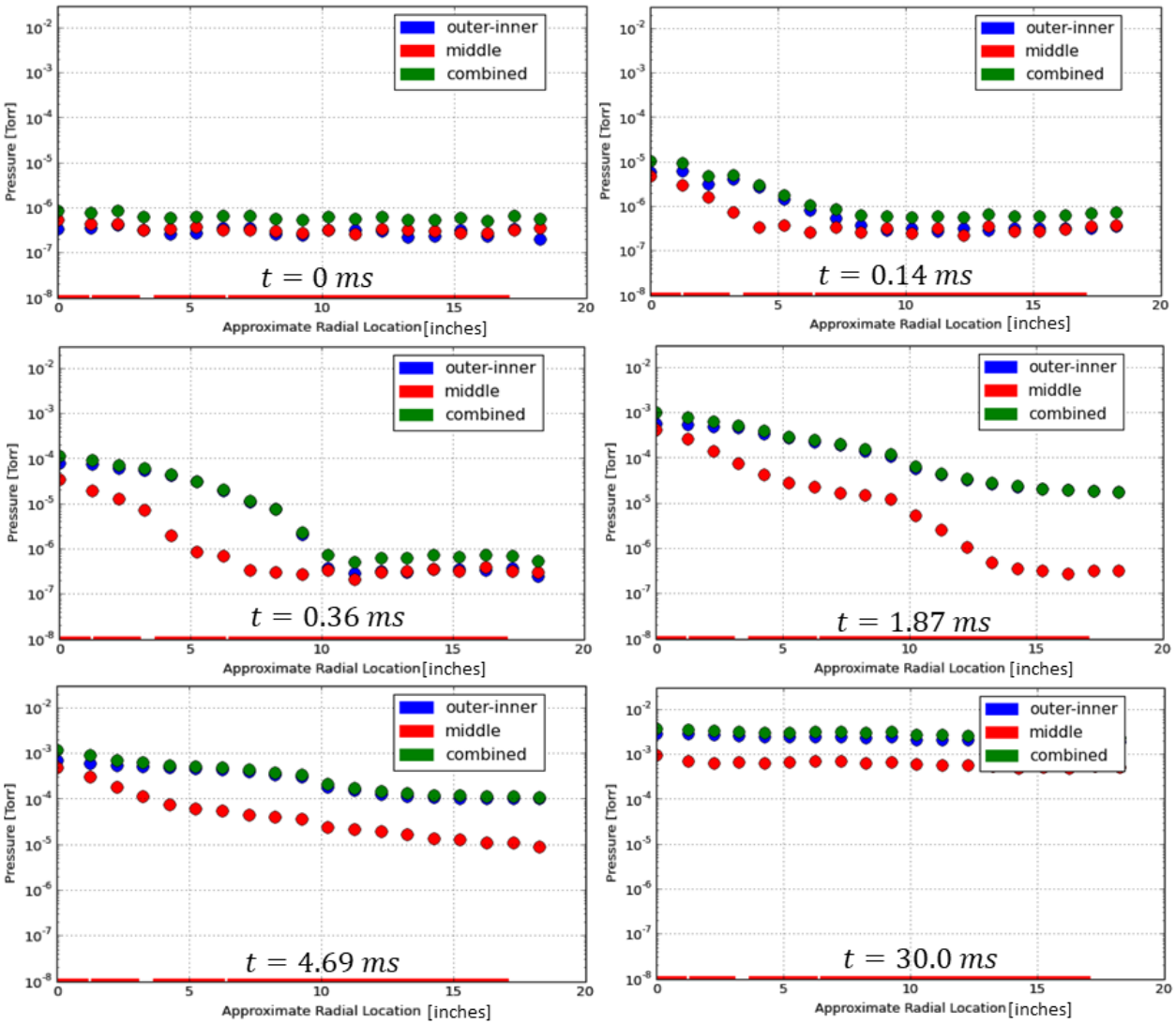


Figure A.7. Evolution of a gas puff in front of the LabJet gun electrodes with a single valve connected to the middle and outer-inner gas injection plenums with a back-line pressure of 100 PSI and gas valve PSU voltage of 1kV. The red lines at the bottom of the plot represent the electrodes with gas slits and high voltage insulating gap shown.

APPENDIX B

DEVELOPMENT OF CUSTOM MULTICHANNEL OPTICAL FIBER BUNDLE

This appendix briefly discusses the development of a cost-effective, custom multichannel optical fiber bundle. The 96-channel, ~10 m long bundle was built using ~1.1 km of 105 μm core glass clad silica multimode optical fibers (FG105LCA) with an NA of 0.22 and an effective wavelength range of (400-2400nm). On the spectrometer side, the fibers are mounted into three 32-channel, 127 μm pitch spacing V-groove chips (PGVG-1321108) which hold the fibers in a linear array. On the plasma side, the fibers are mounted into twelve 8-channel, 127 μm pitch spacing V-groove chips (PGVG-1081105) which also hold the fibers in a linear array. Protective tubing for the fibers is provided by ¼" polyethylene dripline irrigation tubing ($\sim 5m$ for each 8-channel bundle) and ¾" surethane gas line tubing ($\sim 5m$ for all fibers on their way to the spectrometer). Stress is removed from the fibers and the v-groove chips are held in place using custom 3D printed clamps (Figure B.1 and Figure B.2) and a 12-way fiber bundle breakout (Figure B.3) which hold the protective tubing in place by clamping onto ¼" dripline barb connectors. A custom 3D printed barb (Figure B.4) is used for the spectrometer side 32 chip V-groove clamp.

Plasma side collimating optics for each 8-channel bundle was sourced from acrylic bi-aspheric laser diode collimating lenses (191-66010G) with an effective focal length of 4.6mm. These lenses fit perfectly into ¼" gas line compression nuts such that the focus can be tuned by adjusting the distance between the fibers and the lens using the threaded 8-channel v-groove clamp. A ray tracing diagram (Figure B.5) was developed to verify that the lenses provide an adequate depth of field and magnification prior to their purchase. The diagram shows that the 8-indivial fiber lines of sight for each breakout bundle span ~14 cm in the middle of the vacuum chamber ~0.7 m away when the fibers are mounted 0.03mm behind the lens focal point with a resulting magnification of -153.

The steps in the manufacturing process are shown in figure B.6. They proceed as follows:

1. Measure and cut fibers (note that for future arrays significant time and effort can be saved by ordering pre-cut lengths)
2. Thread fibers through protective tubing with barbs already in place.

3. Mount fibers in 8-channel v-groove chips by stripping fiber cladding (most easily accomplished with a soldering iron and NOT the recommended fiber stripper), gluing the end tips, and aligning the fibers into the v-grooves using a 3-axis translation stage and using fiber epoxy to secure the groove lid.
4. Wrap breakout bundles with PTFE tape for extra protection.
5. Clamp 8-channel v-groove chips in the 3D printed clamps.
6. Check continuity of the fibers (after every step), using a laser or other light source.
7. Strip fiber cladding on the spectrometer side
8. Identify corresponding fibers on the spectrometer side using a laser or other light source and arrange in the appropriate order, gluing the end tips to secure in place
9. Align in the 32-channel v-groove chips using the same method as the 8-channel v-groove chips. Secure lid with fiber epoxy.
10. Wrap in PTFE tape for added protection.
11. Clamp in 32-channel v-groove clamp.
12. Cut excess fiber excess (carefully to prevent shattering along the length of the fiber)
13. Polish fibers
14. Epoxy collimating lens to ¼" compression tube nut.

The complete assembly took approximately 3 weeks to complete with one full time worker and part time assistance from the PI. The total cost of the fiber bundle came in at under $2500 and only 2 channels were damaged during the process resulting in a final cost of $\sim\$27/\text{channel}$.

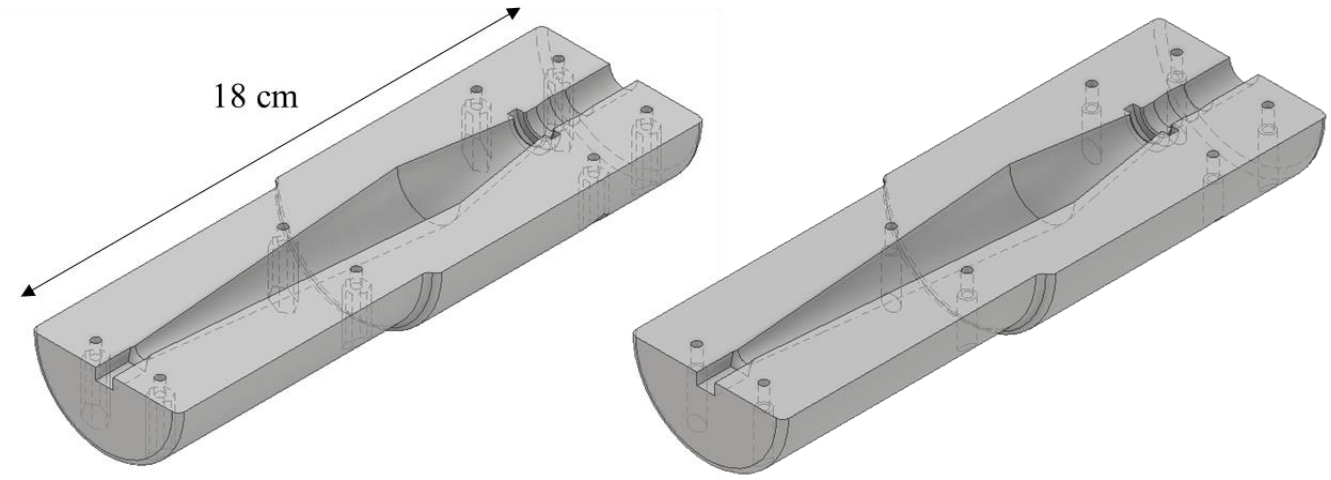


Figure B.1. Custom 3D printed 32-channel V-groove clamp. Bottom of the clamp (left) and top of clamp (right) are virtually identical except for hexagonal holes to house the clamping nuts on the bottom clamp. A larger cavity provides some room for excess fiber between the surethane tubing barb (far side) and the v-groove chips (near side).

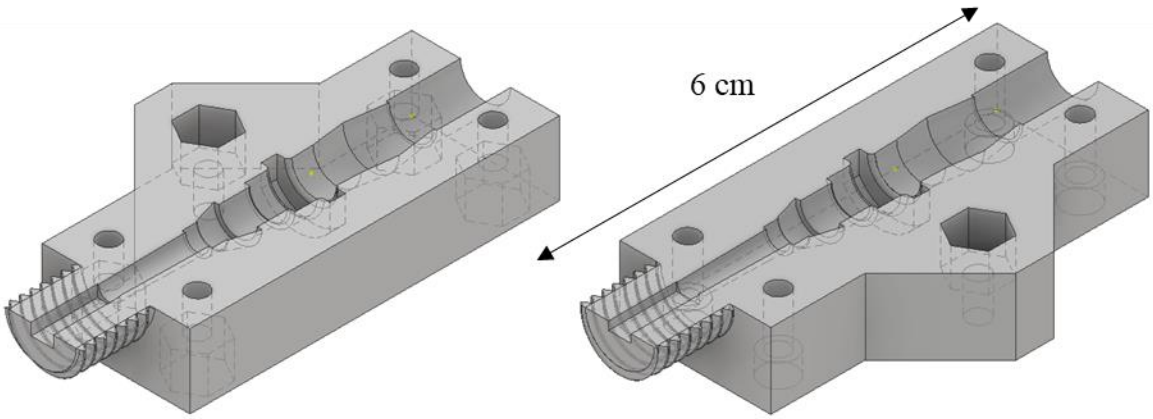


Figure B.2. Custom 3D printed 8 channel V-groove clamp; bottom (left) and top (right). Protective tubing is secured via a ¼" tubing barb in the center of the clamp and the v-groove is secured between the threaded portion of the clamp. Collimating lenses mounted in ¼" gas line compression nuts can be screwed directly onto the clamp to adjust the focus of the fibers.

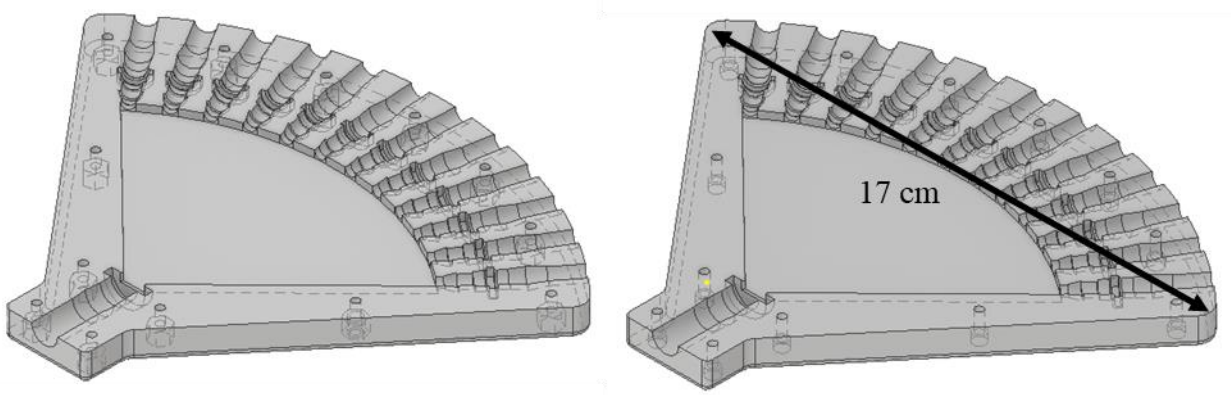


Figure B.3. 12-way fiber bundle breakout clamp; bottom (left) and top (right). Polyethylene tubing for all 12 breakout bundles is secured with ¼" tubing barbs on the far side of the breakout clamp while the ¾" surethane tubing housing all fibers is secured with a custom 3D printed barb on the near side of the clamp (see figure B.4).

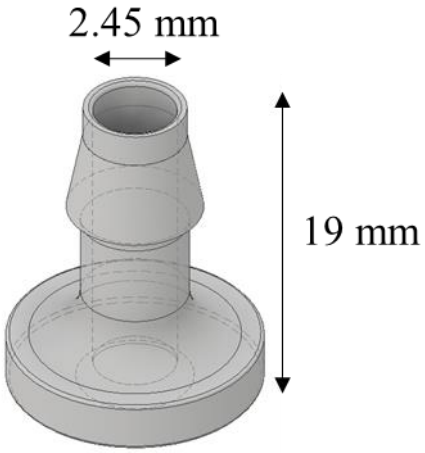


Figure B.4. Custom 3D printed barb used to secure the ¾" surethane tubing in the 12-way fiber bundle breakout clamp and 32 channel v-groove clamp.

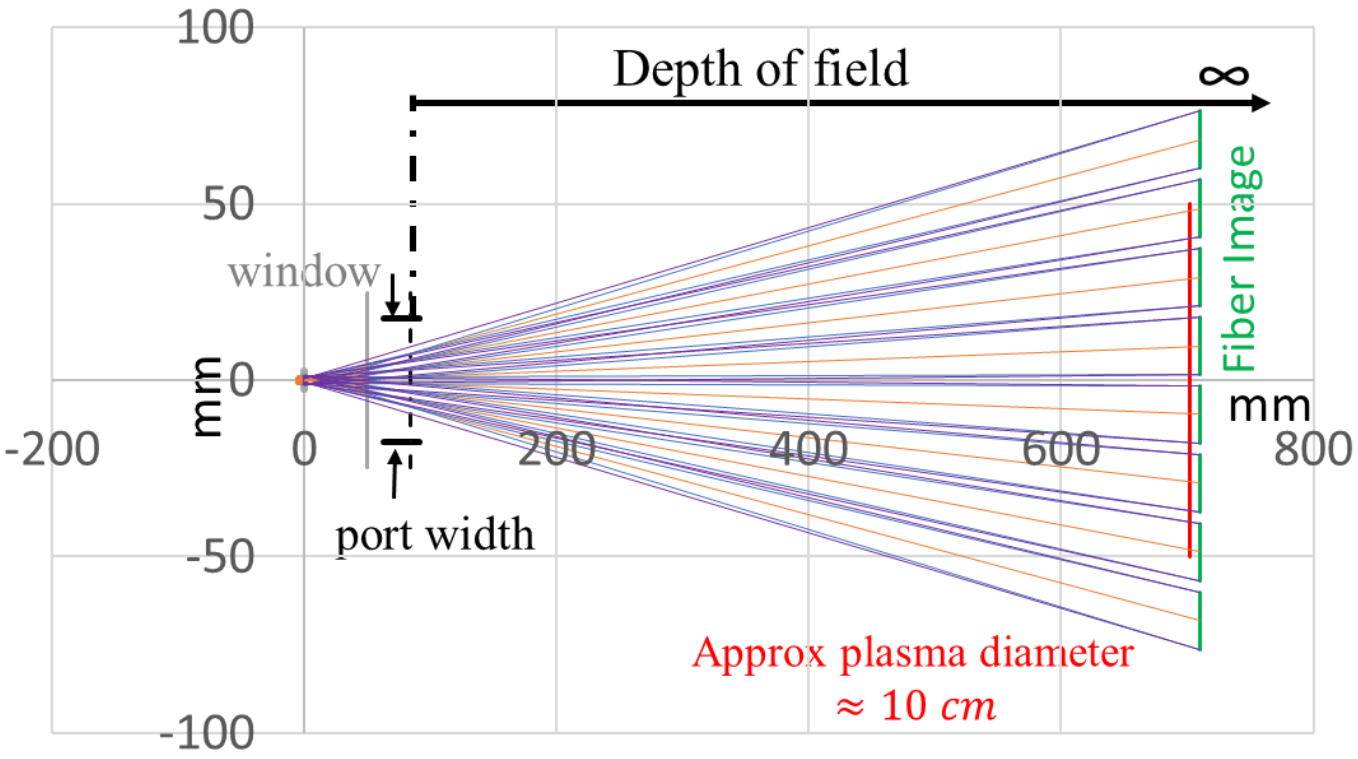


Figure B.5. Ray tracing diagram for plasma side collimating optics. The acrylic bi-aspheric lens has a focal length of 4.6mm. In this diagram, the fibers are mounted 0.03mm behind the focal point resulting in a magnification of -153. The depth of field extends from the vacuum chamber port entrance to infinity.

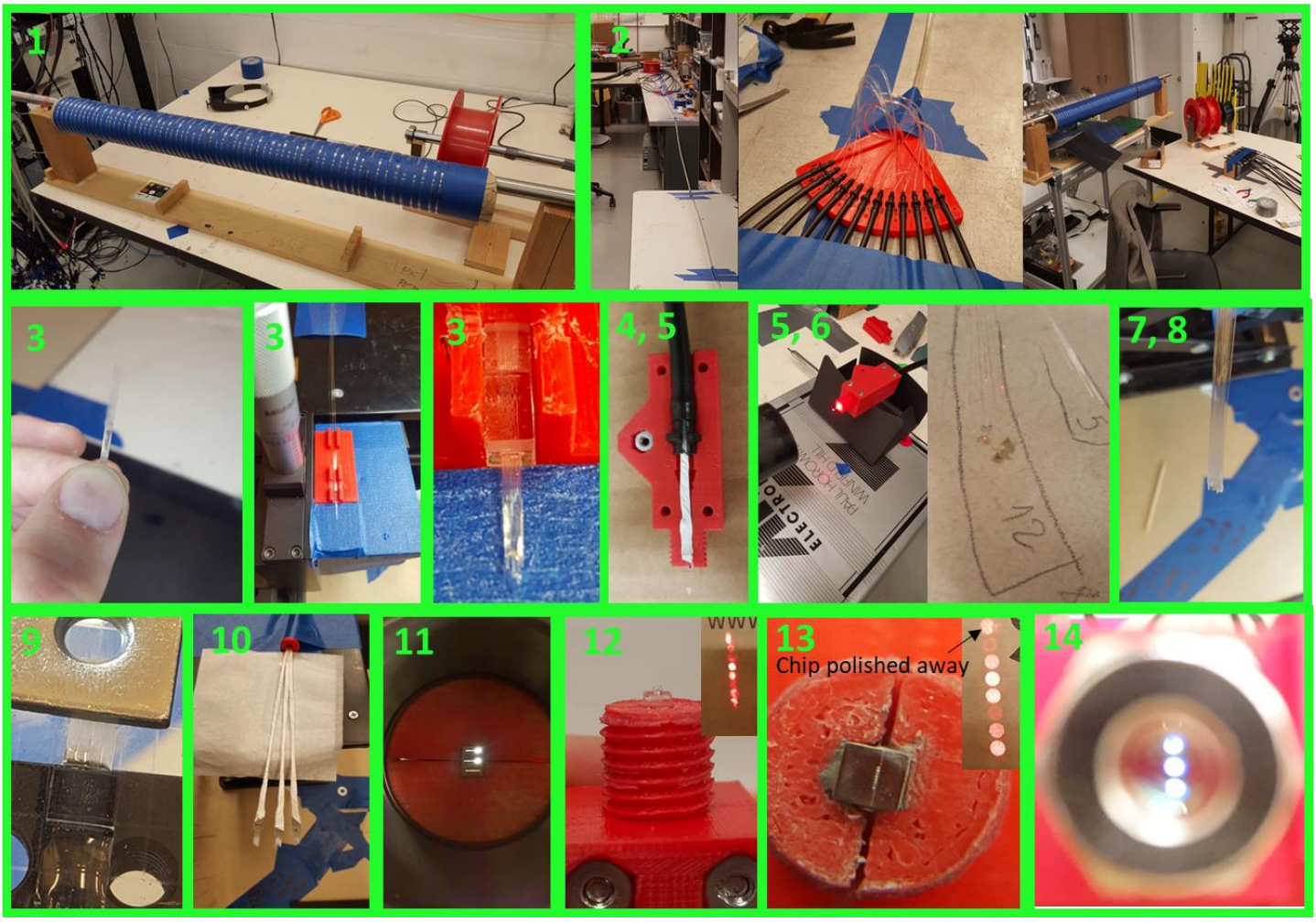


Figure B.6. Photographs taken during the manufacture of the custom fiber bundle. See text for a full description, the numbers of the highlighted photographs correspond the manufacturing steps listed in the text.

APPENDIX C
FLUX TUBE VISUALIZATION CODES

This appendix includes a Mathematica notebook used for the diffuse-core-with skin current kinematic evolution presented in chapter 3. It sets up the relevant vector fields and provides codes to solve for the various streamlines, flux surfaces, and helicity components.

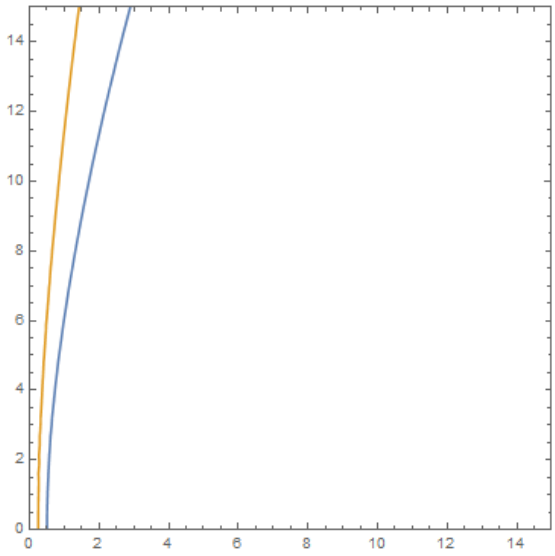
Normalization/Other Constants

```
μ0 = 4 π * 10^-7; (*magnetic permeabilty*)

m0 = 1.6726219 * 10^-27 ; (*Proton mass in KG*)

q0 = 1.60217662 * 10^-19; (*proton charge*)

n0 = 10^20; (*number density in m^-3*)

Z = 1; (*ionization state of ions*)
```

Initialize Fields (Dimensional)

Poloidal Magnetic Field

```
∂hatr[r_] = {0, 1 / r, 0}; (*∇∂*)

l = 15; (*axial length of jet in meters*)

Ivac = 10000; (*bias coil current in Amps*)

avac = 5; (*radius of bias coil in meters*)

α[r_, z_] = Sqrt[avac^2 + r^2 + z^2 - 2 * avac * r];

β[r_, z_] = Sqrt[avac^2 + r^2 + z^2 + 2 * avac * r];

c = μ0 * Ivac / π;

k[r_, z_] = Sqrt[1 - α[r, z]^2 / β[r, z]^2];

A∂[r_, z_] = μ0 * Ivac * avac / (π * Sqrt[avac^2 + r^2 + z^2 + 2 * avac * r]) *
    (((2 - k[r, z]^2) EllipticK[k[r, z]^2] - 2 * EllipticE[k[r, z]^2]) / k[r, z]^2);
 (*toroidal magnetic vector potential in Tesla*meter*)

ψ[r_, z_] = ψ[r_, z_] = If[r == 0, 0, A∂[r, z] * 2 * π * r]; (*magnetic flux in wb,
singularity at r=0 throws 1/0 errors*)

ψjet = 1 * 10^-3; (*define total jet flux in wb*)

ψinner = 2.5 * 10^-4; (*inner surface flux in wb*)

ContourPlot[{ψ[r, z] == ψjet, ψ[r, z] == ψinner}, {r, 0, avac}, {z, 0, l}, PlotRange → {{0, l}, {0, l}},
 AspectRatio → 1]
```




```
afunc = Interpolation[Table[{z, r /. FindRoot[ψ[r, z] == ψjet, {r, 0, 2 + 0 I}]}, {z, -0.1 * l, 1.1 * l, l / 1000}]];
(*jet radius as a function of z (meters)*)

aifunc = Interpolation[Table[{z, r /. FindRoot[ψ[r, z] == ψinner, {r, 0, 2 + 0 I}]}, {z, -0.1 * l, 1.1 * l, l / 1000}]];
(*inner jet surface radius as a function of z (meters)*)

Bpol[r_, z_] =
  {c * z / (2 * α[r, z]^2 * β[r, z] * r) * ((avac^2 + r^2 + z^2) EllipticE[k[r, z]^2] - α[r, z]^2 * EllipticK[k[r, z]^2]),
   0, c / (2 * α[r, z]^2 * β[r, z]) * ((avac^2 - r^2 - z^2) EllipticE[k[r, z]^2] + α[r, z]^2 * EllipticK[k[r, z]^2])};
 (*poloidal magnetic field in Tesla*)
```

Remaining normalization constants that depend on magnetic field strength and jet radius

```
B0 = N[Bpol[10^-100, 0][[3]]] (*normalization field strength in Tesla*)
```

```
0.00125664
```

```
l0 = afunc[0] (*normalize length in m, equal to initial jet radius at z=0*)
```

```
0.502336
```

```
u0 = B0 / Sqrt[μ0 * n0 * m0] (*alfven velocity for normalization field strength and given density, m/s*)
```

```
2740.98
```

```
S0 = q0 * B0 * l0 / (m0 * u0) (*Size parameter*)
```

```
22.0603
```

```
j00 = B0 / (μ0 * l0)
```

```
1990.7
```

Current Profile

```
η[r_, z_] = r / afunc[z];(*self similar parameter*)

ηinterface = 0.6;

Ijet[t_] = t;(*normalized current inside jet*)

jcoreshape[η_] = e^(-2 η^2);(*defines shape of current profile in terms of self similar parameter*)

jcoreshape[(1 - (1 - ηinterface) / 2)]
0.278037

Djcoreshape[η_] = D[jcoreshape[η], η];

jskinshape[η_] =
  InterpolatingPolynomial[{{ηinterface, jcoreshape[ηinterface], Djcoreshape[ηinterface]},
    {1 - (1 - ηinterface) / 2, 1, 0}, {1, 0, 0}}, η];

jη[η_] = Piecewise[{{jcoreshape[η], η ≤ ηinterface}, {jskinshape[η], ηinterface < η ≤ 1}, {0, η > 1}}];

int = Integrate[jη[η] * η, {η, 0, 1}];

j0z[z_, t_] = j0 /. Solve[Ijet[t] == 2 * π * afunc[z]^2 * j0 * int, j0][[1]];

jz[r_, z_, t_] = jη[η[r, z]] * j0z[z, t];

jr[r_, z_, t_] = -1 / r * ∫ r * D[jz[r, z, t], z] dr;
```

Total Fields in Cylindrical Coordinates

```
injet[r_, z_] = If[r ≤ afunc[z], 1, 0];(*sets flows to zero if outside of jet radius*)

Icyl[r_, z_, t_] = 2 * π * Integrate[jz[r1, z, t] * r1, {r1, 0, r}, Assumptions → r ∈ Reals];
(*current as a function of location and total jet current*)

Bcyl[r_, z_, t_] = FullSimplify[Bpol[r, z] + μ0 / (2 π) * Icyl[r, z, t] * θhatr[r]];
(*magnetic field in Tesla*)

jcyl[r_, z_, t_] = {jr[r, z, t], 0, jz[r, z, t]};(*current density in A/m^2*)

uemag = 4 * u0(*magnitude of electron velocity in m/s*)
10963.9

bunitvec[r_, z_, t_] = Bcyl[r, z, t] / Sqrt[Bcyl[r, z, t][[1]]^2 + Bcyl[r, z, t][[2]]^2 + Bcyl[r, z, t][[3]]^2];

uecyl[r_, z_, t_] = uemag * bunitvec[r, z, t] * injet[r, z];(*electron velocity m/s*)

uicyl[r_, z_, t_] = (jcyl[r, z, t] + n0 * q0 * uecyl[r, z, t]) / (n0 * q0 * Z);(*ion velocity in m/s*)

ωecyl[r_, z_, t_] = Curl[uemag * bunitvec[r, z, t], {r, θ, z}, "Cylindrical"] * injet[r, z];

ωicyl[r_, z_, t_] = Curl[(jcyl[r, z, t] + n0 * q0 * uemag * bunitvec[r, z, t]) / (n0 * q0 * Z), {r, θ, z}, "Cylindrical"] *
   injet[r, z];
```

Magnetic Vector Potential

```
k1[r_, rq_, z_, zq_] = 4 * rq * r / ((rq + r)^2 + (zq - z)^2);(*argument for elliptical integral,
rq/zq are integration variables→ flux tube volume*)

azimuthalint1[r_, rq_, z_, zq_] = 4 * EllipticK[k1[r, rq, z, zq]] / Sqrt[(r + rq)^2 + (z - zq)^2];

Ar[r_?NumericQ, z_?NumericQ, t_?NumericQ] :=
  μ0 / (4 π) * NIntegrate[jr[ηq * afunc[zq], zq, t] * ηq * afunc[zq]^2 * azimuthalint1[r, ηq * afunc[zq], z, zq],
    {zq, 0, l}, {ηq, 0, 1}, Method → {Automatic, "SymbolicProcessing" → 0}, AccuracyGoal → 4, PrecisionGoal → 4];

Az[r_?NumericQ, z_?NumericQ, t_?NumericQ] :=
  μ0 / (4 π) * NIntegrate[jz[ηq * afunc[zq], zq, t] * ηq * afunc[zq]^2 * azimuthalint1[r, ηq * afunc[zq], z, zq],
    {zq, 0, l}, {ηq, 0, 1}, Method → {Automatic, "SymbolicProcessing" → 0}, AccuracyGoal → 4, PrecisionGoal → 4];

Acyl[r_, z_, t_] = {Ar[r, z, t], Aϑ[r, z], Az[r, z, t]};
```

Canonical Fields

```
me = 0;

mi = m0;

qe = -q0;

qi = Z * q0;

pecyl[r_, z_, t_] = me * uecyl[r, z, t] + qe * Acyl[r, z, t];

picyl[r_, z_, t_] = mi * uicyl[r, z, t] + qi * Acyl[r, z, t];

Ωecyl[r_, z_, t_] = me * ωecyl[r, z, t] + qe * Bcyl[r, z, t];

Ωicyl[r_, z_, t_] = mi * ωicyl[r, z, t] + qi * Bcyl[r, z, t];
```

Normalizing fields

```
(*input is a normalized position*)

Bnorm[rs_, zs_, t_] = Bcyl[rs * l0, zs * l0, t] / B0;

Anorm[rs_, zs_, t_] = Acyl[rs * l0, zs * l0, t] / (B0 * l0);

uenorm[rs_, zs_, t_] = uecyl[rs * l0, zs * l0, t] / u0;

uinorm[rs_, zs_, t_] = uicyl[rs * l0, zs * l0, t] / u0;

ωenorm[rs_, zs_, t_] = l0 / u0 * ωecyl[rs * l0, zs * l0, t];

ωinorm[rs_, zs_, t_] = l0 / u0 * ωicyl[rs * l0, zs * l0, t];

penorm[rs_, zs_, t_] = me / m0 * uenorm[rs, zs, t] + S0 * qe / q0 * Anorm[rs, zs, t];

Ωenorm[rs_, zs_, t_] = me / m0 * ωenorm[rs, zs, t] + S0 * qe / q0 * Bnorm[rs, zs, t];

pinorm[rs_, zs_, t_] = mi / m0 * uinorm[rs, zs, t] + S0 * qi / q0 * Anorm[rs, zs, t];

Ωinorm[rs_, zs_, t_] = mi / m0 * ωinorm[rs, zs, t] + S0 * qi / q0 * Bnorm[rs, zs, t];

jnorm[rs_, zs_, t_] = jcyl[rs * l0, zs * l0, t] / j00;
```

Converting to cartesian coordinates

```
convert[x_, y_, z_] =
  Transpose[{{x / Sqrt[x^2 + y^2], y / Sqrt[x^2 + y^2], 0}, {-y / Sqrt[x^2 + y^2], x / Sqrt[x^2 + y^2], 0}, {0, 0, 1}}];

B[x_, y_, z_, t_] = convert[x, y, z].Bnorm[Sqrt[x^2 + y^2], z, t];

j[x_, y_, z_, t_] = convert[x, y, z].jnorm[Sqrt[x^2 + y^2], z, t];

ue[x_, y_, z_, t_] = convert[x, y, z].uenorm[Sqrt[x^2 + y^2], z, t];

ui[x_, y_, z_, t_] = convert[x, y, z].uinorm[Sqrt[x^2 + y^2], z, t];

ωe[x_, y_, z_, t_] = convert[x, y, z].ωenorm[Sqrt[x^2 + y^2], z, t];

ωi[x_, y_, z_, t_] = convert[x, y, z].ωinorm[Sqrt[x^2 + y^2], z, t];

A[x_, y_, z_, t_] = convert[x, y, z].Anorm[Sqrt[x^2 + y^2], z, t];

pe[x_, y_, z_, t_] = convert[x, y, z].penorm[Sqrt[x^2 + y^2], z, t];

pi[x_, y_, z_, t_] = convert[x, y, z].pinorm[Sqrt[x^2 + y^2], z, t];

Ωe[x_, y_, z_, t_] = convert[x, y, z].Ωenorm[Sqrt[x^2 + y^2], z, t];

Ωi[x_, y_, z_, t_] = convert[x, y, z].Ωinorm[Sqrt[x^2 + y^2], z, t];

ucm[x_, y_, z_, t_] = (mi / m0 * ui[x, y, z, t] + me / m0 * ue[x, y, z, t]) / (mi / m0 + me / m0);
```

magnetic flux contours

```
c = RevolutionPlot3D[{afunc[z] / l0, z / l0}, {z, 0, l}, {ϑ, π / 15, 2 π + π / 15}, Mesh → {0, {π / 15, 1.2 π}},
   PlotStyle → {{FaceForm[None], EdgeForm[{None}]}, {Opacity[0.5], FaceForm[White], EdgeForm[None]}},
   Lighting → "Neutral",
   PlotRange → {{(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {0, l / l0}},
   Axes → False, Boxed → False];

ccut = RevolutionPlot3D[{afunc[z] / l0, z / l0}, {z, 0, l}, {ϑ, -π / 4, 3 π / 2}, Mesh → {0, {π / 15, 1.2 π}},
   PlotStyle → {{FaceForm[None], EdgeForm[{None}]}, {Opacity[0.5], FaceForm[White], EdgeForm[None]}},
   Lighting → "Neutral",
   PlotRange → {{(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {0, l / l0}},
   Axes → False, Boxed → False];

ci = RevolutionPlot3D[{aifunc[z] / l0, z / l0}, {z, 0, l}, {ϑ, π / 15, 2 π + π / 15}, Mesh → {0, {π / 15, 1.2 π}},
   PlotStyle → {{FaceForm[None], EdgeForm[{None}]}, {Opacity[0.8], FaceForm[White], EdgeForm[None]}},
   Lighting → "Neutral",
   PlotRange → {{(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {0, l / l0}},
   Axes → False, Boxed → False];
```

Canonical Vorticity flux contours

```
Ψplot[r_, z_, t_] := Module[{flux, radius, r1, table, aconfunc, plot},
  flux = 2 * π * r * (mi * uicyl[r, z, t][[2]] + qi * Aθ[r, z]);
  radius[z1_] := r1 /. FindRoot[2 * π * r1 * (mi * uicyl[r1, z1, t][[2]] + qi * Aθ[r1, z1]) == flux, {r1, 0.1, afunc[l] - 3}];
  table = Table[{z1, radius[z1]}, {z1, z, l, l / 1000}];
  aconfunc = Interpolation[table];
  plot = RevolutionPlot3D[{aconfunc[z1] / l0, z1 / l0}, {z1, z, l}, {θ, 1.1 π, 2 π + 1.1 π}, Mesh → {0, {1.1 π, 2.2 π}},
    PlotStyle → {{FaceForm[None], EdgeForm[{None}]}, {Opacity[0.5], FaceForm[Black], EdgeForm[None]}},
    Lighting → "Neutral",
    PlotRange → {{(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {0, l / l0}},
    Axes → False, Boxed → False]
  (*note this takes non normalized inputs but outputs the normalized plot → for example input r=
  afunc[0] and plot will start w/ flux tube foot at 1= afunc[0]/l0*)

Ψplotcut[r_, z_, t_] := Module[{flux, radius, r1, table, aconfunc, plot},
  flux = 2 * π * r * (mi * uicyl[r, z, t][[2]] + qi * Aθ[r, z]);
  radius[z1_] := r1 /. FindRoot[2 * π * r1 * (mi * uicyl[r1, z1, t][[2]] + qi * Aθ[r1, z1]) == flux, {r1, 0.1, afunc[l] - 3}];
  table = Table[{z1, radius[z1]}, {z1, z, l, l / 1000}];
  aconfunc = Interpolation[table];
  plot = RevolutionPlot3D[{aconfunc[z1] / l0, z1 / l0}, {z1, z, l}, {θ, -π / 4 + π, 3 π / 2 + π},
    Mesh → {0, {1.1 π, 2.2 π}},
    PlotStyle → {{FaceForm[None], EdgeForm[{None}]}, {Opacity[0.5], FaceForm[Black], EdgeForm[None]}},
    Lighting → "Neutral",
    PlotRange → {{(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {(-1.2 * afunc[l]) / l0, (1.2 * afunc[l]) / l0}, {0, l / l0}},
    Axes → False, Boxed → False]
  (*note this takes non normalized inputs but outputs the normalized plot → for example input r=
  afunc[0] and plot will start w/ flux tube foot at 1= afunc[0]/l0*)
```

Streamline code

```
streamup[f_, time_, x0_, y0_, z0_, smax_, color_, dashing_, thickness_, accuracy_, maxstep_, discontinuity_] :=
 Module[{atboundary, p, eqs, ic, solution, xans, yans, zans, send, pplot},
  atboundary = If[z[s] == (l + 5) / l0 || z[s] == 0, True, False];
  p[s_] := {x[s], y[s], z[s]};
  eqs := Thread[p'[s] == f[x[s], y[s], z[s], time]];
  ic := {Thread[p[0] == {x0, y0, z0}]};
  solution = NDSolve[{eqs, ic, WhenEvent[z[s] == (1.1 * l) / l0, "StopIntegration"], WhenEvent[z[s] == 0, "StopIntegration"]},
    {x, y, z}, {s, 0, smax}, AccuracyGoal → accuracy, MaxStepSize → maxstep, PrecisionGoal → 4,
    Method → {"DiscontinuityProcessing" → discontinuity}];
  xans = x /. solution[[1, 1]];
  yans = y /. solution[[1, 2]];
  zans = z /. solution[[1, 3]];
  send = InterpolatingFunctionDomain[zans][[1, 2]];
  pplot = ParametricPlot3D[{xans[s], yans[s], zans[s]}, {s, 0, send},
    PlotStyle → {Thickness[thickness], color, Dashing → dashing},
    PlotRange → {{-1.2 afunc[l] / l0, 1.2 afunc[l] / l0}, {-1.2 afunc[l] / l0, 1.2 afunc[l] / l0}, {-2, 1.1 * l / l0}},
    Axes → False, Boxed → False];
  pplot /. Line[s_] :> {Arrowheads[Table[.02, {5}]], Arrow[s]}
 ]
```

```
streamdown[f_, time_, x0_, y0_, z0_, smin_, color_, dashing_, thickness_, accuracy_, maxstep_, discontinuity_] :=
 Module[{p, eqs, ic, solution, xans, yans, zans, send, pplot},
  p[s_] := {x[s], y[s], z[s]};
  eqs := Thread[p'[s] == f[x[s], y[s], z[s], time]];
  ic := {Thread[p[0] == {x0, y0, z0}]};
  solution = NDSolve[{eqs, ic, WhenEvent[z[s] == 0, "StopIntegration"], WhenEvent[z[s] == 1.1 * l / l0, "StopIntegration"]},
    {x, y, z}, {s, 0, smin}, AccuracyGoal → accuracy, MaxStepSize → maxstep, PrecisionGoal → 4,
    Method → {"DiscontinuityProcessing" → discontinuity}];
  xans = x /. solution[[1, 1]];
  yans = y /. solution[[1, 2]];
  zans = z /. solution[[1, 3]];
  send = InterpolatingFunctionDomain[zans][[1, 1]];
  pplot = ParametricPlot3D[{xans[s], yans[s], zans[s]}, {s, 0, send},
    PlotStyle → {Thickness[thickness], color, Dashing -> dashing},
    PlotRange → {{-1.2 afunc[l] / l0, 1.2 afunc[l] / l0}, {-1.2 afunc[l] / l0, 1.2 afunc[l] / l0}, {-2, 1.1 * l / l0}},
    Axes → False, Boxed → False];
  pplot /. Line[s_] :> {Arrowheads[Table[.02, {5}]], Arrow[s]}
 ]

referenceline =
  Graphics3D[{{Dashing → {.1, .006}, Line[{{0, 0, 0}, {0, 0, l / l0}}]},
    {Thickness[Large], Line[{{-0.01, 0, 0}, {0.01, 0, 0}}]}, {Thickness[Large], Line[{{0, -.01, 0}, {0, 0.01, 0}}]},
    {Thickness[Large], Line[{{-0.2, 0, l / l0}, {0.2, 0, l / l0}}]},
    {Thickness[Large], Line[{{0, -.2, l / l0}, {0, 0.2, l / l0}}]}}];

Streamline[f_, time_, x0_, y0_, z0_, smin_, smax_, color_, dashing_, thickness_, refs_, accuracy_, maxstep_,
  discontinuity_] :=
 Show[streamup[f, time, x0, y0, z0, smax, color, dashing, thickness, accuracy, maxstep, discontinuity],
  streamdown[f, time, x0, y0, z0, smin, color, dashing, thickness, accuracy, maxstep, discontinuity], referenceline]

vector[field_, tv_, xv_, yv_, zv_, multiplier_, dashing_, thickness_, arrowsize_] :=
  Graphics3D[{{Dashing → dashing, Arrowheads[arrowsize], Thickness[thickness],
     Arrow[{{xv, yv, zv}, {xv, yv, zv} + multiplier * field[xv, yv, zv, tv]}]}}];
```

Kinematic Evolution

Magnetic Field and Current Density

```
Bstream[t_, bvec_, jvec_] :=
 Show[Streamline[B, t, 0, (-afunc[0.01]) / l0, .01 / l0, -∞, 36, Black, None, .002, 1, Automatic, Automatic, True],
  Streamline[ucm, t, 0, (-.999 afunc[0.01]) / l0, .01 / l0, -∞, 8, Black, {0.05, 0.02}, .01, 1, Automatic, Automatic, True],
  Streamline[B, t, 0, (-aifunc[0.01]) / l0, .01 / l0, -∞, 300, Black, None, .002, 1, Automatic, Automatic, True],
  Streamline[j, t, 0, (-.999 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.03, .005}, .006, 1, Automatic, Automatic, True],
  Streamline[j, t, 0, (-aifunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.03, .005}, .006, 1, Automatic, Automatic, True],
  Streamline[ucm, t, 0, (-.999 aifunc[0.01]) / l0, .01 / l0, -∞, 18, Black, {0.05, 0.02}, .01, 1, Automatic, Automatic, True],
  vector[B, t, 0, (-afunc[0.01]) / l0, 0.01 / l0, bvec, None, .01, .04],
  vector[j, t, 0, (-afunc[0.01]) / l0, 0.01 / l0, jvec, {.03, .005}, .01, .04], ccut, ci, ImageSize → Large]
```

Flows

```
Flows[t_, uevec_, uivec_] :=
 Show[Streamline[ue, t, 0, (-.999 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, None, .002, 1, Automatic, Automatic, True],
  Streamline[ue, t, 0, (-.999 aifunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, None, .002, 1, Automatic, Automatic, True],
  Streamline[ui, t, 0, (-.999 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {0.05, 0.01}, .01, 1, Automatic, Automatic, True],
  Streamline[ui, t, 0, (-.999 aifunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {0.05, 0.01}, .01, 1, Automatic, Automatic, True],
  vector[ue, t, 0, (-.999 afunc[0.01]) / l0, 0.01 / l0, uevec, None, .004, .02],
  vector[ui, t, 0, (-.999 afunc[0.01]) / l0, 0.01 / l0, uivec, {.03, .005}, .01, .04], ccut, ci, ImageSize → Large]
```

Canonical Momentum

```
Pstream[t_, pevec_, pivec_] :=
 Show[Streamline[pe, t, 0, (-.99 afunc[0.01]) / l0, .01 / l0, -100, 100, Black, None, .003, 1, 5, Automatic, False],
  Streamline[pe, t, 0, (-.99 aifunc[0.01]) / l0, .01 / l0, -100, 100, Black, None, .003, 1, 5, Automatic, False],
  Streamline[pi, t, 0, (-.99 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.05, .01}, .01, 1, 5, Automatic, False],
  Streamline[pi, t, 0, (-.99 aifunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.05, .01}, .01, 1, 5, Automatic, False],
  vector[pe, t, 0, (-.99 afunc[0.01]) / l0, 0.01 / l0, pevec, None, .005, .03],
  vector[pi, t, 0, (-.99 afunc[0.01]) / l0, 0.01 / l0, pivec, {.03, .005}, .01, .04], ccut, ci, ImageSize → Large]
```

Canonical Vorticity

```
Ωstream[t_, Ωevec_, Ωivec_] :=
 Show[Streamline[Ωe, t, 0, (-.99 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, None, .003, 1, 5, Automatic, False],
  (*Streamline[Ωe,t,0,(-.99aifunc[0.01])/l0,.01/l0,-∞,∞,Black,None,.003,1,5,Automatic,False],*)
  Streamline[Ωi, t, 0, (-.99 afunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.05, .01}, .01, 1, 5, Automatic, False],
  Streamline[Ωi, t, 0, (-.99 aifunc[0.01]) / l0, .01 / l0, -∞, ∞, Black, {.05, .01}, .01, 1, 5, Automatic, False],
  vector[Ωe, t, 0, (-.99 afunc[0.01]) / l0, 0.01 / l0, Ωevec, None, .005, .03],
  vector[Ωi, t, 0, (-.99 afunc[0.01]) / l0, 0.01 / l0, Ωivec, {.05, .01}, .01, .04], ccut,
  Ψplotcut[(.99 afunc[0.01]), .01, t], Ψplot[(.99 aifunc[0.01]), .01, t], ImageSize → Large]
```

Normalized Relative Canonical Helicity

```
(*INPUT VALUES ARE THE NORMALIZED position VALUES!!!*)

Needs["NDSolve`FEM`"]

anorm[z_] = afunc[z * l0] / l0; (*anorm(znorm)=a(znorm*l0)/l0*)

lnorm = l / l0;

tanϑ[z_] = Bpol[afunc[z * l0], z * l0][[1]] / Bpol[afunc[z * l0], z * l0][[3]]; (*gives Tan[ϑ]  as a function of Z,
where ϑ is the angle between Bz and Br*)

ϑ[z_] = ArcTan[tanϑ[z]];

nvec[z_] := {Cos[ϑ[z]], 0, -Sin[ϑ[z]]};

region = ImplicitRegion[0 ≤ z ≤ lnorm && 0 ≤ r ≤ anorm[z], {r, z}];

mesh = ToElementMesh[region, "BoundaryMeshGenerator" → "Continuation", "MaxBoundaryCellMeasure" → 0.1,
  MaxCellMeasure → 0.3]

ElementMesh[{{0., 5.75791}, {0., 29.8605}}, {TriangleElement[<3213>]}]

mesh["Wireframe"];

Biref[r_, z_] = -Bpol[r * l0, z * l0] / B0;

Airef[r_, z_] = {0, -Aϑ[r * l0, z * l0] / (B0 * l0), 0};
```

```
referencepotential[current_] := Module[{sideBC, bottomBC, topBC, laplacian, solution},
  sideBC = Interpolation[Table[{z, Dot[ωinorm[anorm[z], z, current], nvec[z]]}, {z, 0, l/l0, 0.5}]];
  bottomBC =
   Interpolation[
    Table[{r, If[r == 0, (Dot[ωinorm[10^-10, 0, current], {0, 0, -1}]), (Dot[ωinorm[r, 0, current], {0, 0, -1}])]},
     {r, 0, anorm[0], 0.005}]];
  topBC =
   Interpolation[
    Table[{r, If[r == 0, (Dot[ωinorm[10^-10, l/l0, current], {0, 0, 1}]), (Dot[ωinorm[r, l/l0, current], {0, 0, 1}])]},
     {r, 0, anorm[l/l0], 0.05}]];
  laplacian = D[ϕ[r, z], {r, 2}] + D[ϕ[r, z], r]/r + D[ϕ[r, z], {z, 2}];
  solution =
   NDSolveValue[
    {laplacian == NeumannValue[sideBC[z], r == anorm[z]] + NeumannValue[bottomBC[r], z == 0] +
       NeumannValue[topBC[r], z == l/l0]}, ϕ, {r, z} ∈ mesh]
 ]

IonHelicityComponents[current_?NumericQ] :=
 Module[{ϕt, rderiv, zderiv, ωiref, uiref, piref, Ωiref, integrand1, integrand2, integrand3, integrand4,
   Magrel, Flowrel, Xrel, TotalHelicity},
  ϕt = referencepotential[current];
  rderiv = Derivative[1, 0][ϕt];
  zderiv = Derivative[0, 1][ϕt];
  ωiref[r_?NumericQ, z_?NumericQ] = {rderiv[r, z], 0, zderiv[r, z]};
  uiref[r_?NumericQ, z_?NumericQ] :=
   {0,
    2 π * NIntegrate[zderiv[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 0, r/anorm[z]},
       Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 4, AccuracyGoal → 4]/(2 π * r), 0};
  piref[r_?NumericQ, z_?NumericQ] := mi/m0 * uiref[r, z] + S0 * qi/q0 * Airef[r, z];
  Ωiref[r_?NumericQ, z_?NumericQ] := mi/m0 * ωiref[r, z] + S0 * qi/q0 * Biref[r, z];
  integrand1[r_?NumericQ, z_?NumericQ] := (pinorm[r, z, current] - piref[r, z]).(Ωinorm[r, z, current] + Ωiref[r, z]);
  integrand2[r_?NumericQ, z_?NumericQ] := (Bnorm[r, z, current] + Biref[r, z]).(Anorm[r, z, current] - Airef[r, z]);
  integrand3[r_?NumericQ, z_?NumericQ] := (uinorm[r, z, current] - uiref[r, z]).(ωinorm[r, z, current] + ωiref[r, z]);
  integrand4[r_?NumericQ, z_?NumericQ] :=
   (uinorm[r, z, current] - uiref[r, z]).(Bnorm[r, z, current] + Biref[r, z]) +
    (ωinorm[r, z, current] + ωiref[r, z]).(Anorm[r, z, current] - Airef[r, z]);
  TotalHelicity = 2 π * NIntegrate[integrand1[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 2, AccuracyGoal → 3];
  Magrel = 2 π * NIntegrate[integrand2[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 2, AccuracyGoal → 3];
  Flowrel = 2 π * NIntegrate[integrand3[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 2, AccuracyGoal → 3];
  Xrel = 2 π * NIntegrate[integrand4[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 2, AccuracyGoal → 3];
  {TotalHelicity, (qi/q0)^2 * S0^2 Magrel, (mi/m0)^2 Flowrel, (mi/m0) * (qi/q0) * S0 * Xrel}
 ]


MagHelicity[current_?NumericQ] := Module[{integrand2, Magrel},
  integrand2[r_?NumericQ, z_?NumericQ] := (Bnorm[r, z, current] + Biref[r, z]).(Anorm[r, z, current] - Airef[r, z]);
  Magrel = 2 π * NIntegrate[integrand2[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 3];
  (qi/q0)^2 * S0^2 Magrel
 ]
```

```
FlowHelicity[current_?NumericQ] := Module[{ϕt, rderiv, zderiv, ωiref, uiref, integrand3, Flowrel},
  ϕt = referencepotential[current];
  rderiv = Derivative[1, 0][ϕt];
  zderiv = Derivative[0, 1][ϕt];
  ωiref[r_?NumericQ, z_?NumericQ] = {rderiv[r, z], 0, zderiv[r, z]};
  uiref[r_?NumericQ, z_?NumericQ] :=
   {0,
    2 π * NIntegrate[zderiv[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 0, r / anorm[z]},
       Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 4, AccuracyGoal → 4] / (2 π * r), 0};
  integrand3[r_?NumericQ, z_?NumericQ] := (uinorm[r, z, current] - uiref[r, z]).(ωinorm[r, z, current] + ωiref[r, z]);
  Flowrel = 2 π * NIntegrate[integrand3[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 1];
  (mi / m0)^2 Flowrel
 ](*Takes a long time at low currents... solve backwards by subtracting onther components from the total*)

CrossHelicity[current_?NumericQ] := Module[{ϕt, rderiv, zderiv, ωiref, uiref, integrand4, Xrel},
  ϕt = referencepotential[current];
  rderiv = Derivative[1, 0][ϕt];
  zderiv = Derivative[0, 1][ϕt];
  ωiref[r_?NumericQ, z_?NumericQ] = {rderiv[r, z], 0, zderiv[r, z]};
  uiref[r_?NumericQ, z_?NumericQ] :=
   {0,
    2 π * NIntegrate[zderiv[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 0, r / anorm[z]},
       Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 4, AccuracyGoal → 4] / (2 π * r), 0};
  integrand4[r_?NumericQ, z_?NumericQ] :=
   (uinorm[r, z, current] - uiref[r, z]).(Bnorm[r, z, current] + Biref[r, z]) +
    (ωinorm[r, z, current] + ωiref[r, z]).(Anorm[r, z, current] - Airef[r, z]);
  Xrel = 2 π * NIntegrate[integrand4[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 1];
  (mi / m0) * (qi / q0) * S0 * Xrel
 ](*takes a long time at high currents... solve backwards by subtracting other components from the total*)

TotalHelicity[current_?NumericQ] := Module[{ϕt, rderiv, zderiv, ωiref, uiref, piref, Ωiref, integrand1, TotalHelicity},
  ϕt = referencepotential[current];
  rderiv = Derivative[1, 0][ϕt];
  zderiv = Derivative[0, 1][ϕt];
  ωiref[r_?NumericQ, z_?NumericQ] = {rderiv[r, z], 0, zderiv[r, z]};
  uiref[r_?NumericQ, z_?NumericQ] :=
   {0,
    2 π * NIntegrate[zderiv[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 0, r / anorm[z]},
       Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 4, AccuracyGoal → 4] / (2 π * r), 0};
  piref[r_?NumericQ, z_?NumericQ] := mi / m0 * uiref[r, z] + S0 * qi / q0 * Airef[r, z];
  Ωiref[r_?NumericQ, z_?NumericQ] := mi / m0 * ωiref[r, z] + S0 * qi / q0 * Biref[r, z];
  integrand1[r_?NumericQ, z_?NumericQ] := (pinorm[r, z, current] - piref[r, z]).(Ωinorm[r, z, current] + Ωiref[r, z]);
  TotalHelicity = 2 π * NIntegrate[integrand1[ηq * anorm[z], z] * ηq * anorm[z]^2, {ηq, 10^-5, 1}, {z, 0, lnorm},
     Method → {Automatic, "SymbolicProcessing" → 0}, PrecisionGoal → 1]
 ]
```

Appendix D
STABILITY SPACE CODES

This appendix contains codes from a Mathematica notebook used to plot the stability space evolution of the jets in the MoCHI experiment. The codes primarily import camera images and track the intensity profiles to determine jet length and radius.

The following code was used for figure 5.23. It imports a camera image and fits a Gaussian to the image intensity profile along the selected line with end points (x1,y1), (x2,y2). The background intensity is mapped with a quadratic and removed.

```
Needs["ErrorBarPlots`"]

ImageAnalyze[img_, x1_, y1_, x2_, y2_] :=
 Module[{image, data, Δx, Δy, l, intensityprofile, background, backgroundfit, shifted, gaussian, fit, sol,
   max, sides, p1, p2, fwhm},
  image = ImageAdjust[Import[img], {0.5, 9, 0.5}];
  data = Import[img, "Data"];
  Δx = x2 - x1;
  Δy = y2 - y1;
  l = Sqrt[Δx^2 + Δy^2];
  intensityprofile = Table[{i, data[[(768 - y1) - Round[Δy * i / l], x1 + Round[Δx * i / l]]]}, {i, 0, Round[l]}];

  background = Drop[intensityprofile, {Round[l / 3], Round[2 * l / 3]}];
  (*defines background as outer portion of profile*)
  backgroundfit = NonlinearModelFit[background, a * (x - b)^2 + c, {a, b, c}, x];
  (*fits quadratic to background*)

  shifted = Table[{i, intensityprofile[[i, 2]] - backgroundfit[i]}, {i, 1, Round[l]}];

  gaussian[x_] = amp * Evaluate[PDF[NormalDistribution[x0, σ], x]];
  fit = FindFit[shifted[[Round[l / 3] ;; Round[2 * l / 3]]], gaussian[x], {{amp, 200}, {x0, l / 2}, {σ, l / 4}}, x];

  sol[x_] = gaussian[x] /. fit;
  max = FindMaxValue[sol[x], {x, 100}];
  sides = Quiet@Solve[sol[x] == max / 2, x];
  p1 = x /. sides[[1]];
  p2 = x /. sides[[2]];

  fwhm = Abs[p2 - p1];


  {GraphicsGrid[{{Show[image, Graphics[{Red, Thick, Line[{{x1, y1}, {x2, y2}}]}]]},
      Show[ListPlot[{intensityprofile, background, shifted}, PlotRange → All, PlotLabel → fwhm],
       Plot[{sol[x] + backgroundfit[x], backgroundfit[x], sol[x]}, {x, 0, l}, PlotRange → All],
       Graphics[{Red, Thick, Line[{{p1, sol[p1]}, {p2, sol[p2]}}]}]]]}], fwhm}

  (*GraphicsGrid[
   {{Show[image,Graphics[{White,Thick,Line[{{x1,y1},{x2,y2}}]}]],
     Show[ListPlot[intensityprofile,PlotRange→All,BaseStyle→{Black,FontSize→30},PlotTheme→"Monochrome",Frame→True],
      Plot[{sol[x]+backgroundfit[x],backgroundfit[x],sol[x]},{x,0,l},PlotRange→Full,PlotTheme→"Monochrome"],
      Graphics[{Gray,Thickness[.005],Line[{{p1,sol[p1]},{p2,sol[p2]}}]}]]}]*)(*publication quality plot*)
]
```

The following code determines the length in pixels of the jet based on the intensity profile along a line with end points (x1,y1), (x2,y2). The last 10% of the line defines the background level.

```
length[img_, x1_, y1_, x2_, y2_] :=
 Module[{image, data, Δx, Δy, l, intensityprofile, background, locations},
  image = ImageAdjust[Import[img], {0.05, 9, 0.5}];
  data = Import[img, "Data"];
  Δx = x2 - x1;
  Δy = y2 - y1;
  l = Sqrt[Δx^2 + Δy^2];
  intensityprofile = Table[{i, data[[(768 - y1) - Round[Δy * i / l], x1 + Round[Δx * i / l]]]}, {i, 0, Round[l]}];
  background = N[Mean[intensityprofile[[Round[0.9 * l] ;; Round[l], 2]]]];

  locations = Position[intensityprofile[[ ;; , 2]], _?(# < background * 1.1 &)];

  {GraphicsGrid[{{Show[image, Graphics[{Red, Thick, Line[{{x1, y1}, {x2, y2}}]}]],
      Show[ListPlot[intensityprofile, PlotLabel → intensityprofile[[locations[[7]], 1]],
        BaseStyle → {Black, FontSize → 50}, Frame → True, PlotStyle → Directive[PointSize[.02]]],
       Plot[{background, 1.1 * background}, {x, 0, l}, BaseStyle → {Black, FontSize → 50}, PlotTheme → "Monochrome",
        Frame → True],
       Graphics[{Red, Thick, Line[{{intensityprofile[[locations[[7]], 1]][[1]], 0},
           {intensityprofile[[locations[[7]], 1]][[1]], 800}}]}]]}], intensityprofile[[locations[[7]], 1]][[1]]}
 ]
```

APPENDIX E

IDS CODES

The following Mathematica notebook was used for the analysis of H_α line emission, in this case for shot 7143.

Import files and define shot settings

```
Needs["ErrorBarPlots`"]

ref =
  Import[
   "E:\\All Documents\\AAHYDRA FILES\\Spectroscopy Images for analysis\\Fiber bundle data\\Calibration
     images\\656.3(hydrogen-656.3) 2017 May 23 15_23_25-Frame-1.csv"];
frame1 =
  Import[
   "E:\\All Documents\\AAHYDRA FILES\\Spectroscopy Images for analysis\\Fiber bundle data\\OP 28\\2017 May 24
     12_59_12 007143-Frame-1.csv"];
frame2 =
  Import[
   "E:\\All Documents\\AAHYDRA FILES\\Spectroscopy Images for analysis\\Fiber bundle data\\OP 28\\2017 May 24
     12_59_12 007143-Frame-2.csv"];

referenceimage =
  ImageAdjust[
   Import[
    "E:\\All Documents\\AAHYDRA FILES\\Spectroscopy Images for analysis\\Fiber bundle data\\Calibration
      images\\656.3(hydrogen-656.3) 2017 May 23 15_23_25-Roi-1 .tif"], {2, 2, .5}];
 (*import .tiff reference image file*)
```

Dispersion relation and doppler resolution

```
σλ = 0.05 * 10^-9;

σλavg = 1 / 2 * Sqrt[2 (σλ)^2];

R[λ_] := Module[{dx, L, d, m, α, c},
  dx = 12.8 / 10^6;
  L = 1;
  d = 1 / (2400 * 10^3);
  m = 1;
  α = 5.25 * π / 180;
  c = 3 * 10^8;

  (dx / (2 L) * (Sqrt[(2 * d * Cos[α] / m)^2 - (λ * 10^-9)^2] + (λ * 10^-9) * Tan[α])) * 10^9
 ](*linear dispersion relation of the monochromater in nm/px*)
```

```
σR[λ_] := Module[{dx, L, d, m, α, c, c1, c2, c3},
  dx = 12.8 / 10^6;
  L = 1;
  d = 1 / (2400 * 10^3);
  m = 1;
  α = 5.25 * π / 180;
  c = 3 * 10^8;

  c1 = dx / 2 * L;
  c2 = (2 * d * Cos[α] / m) ^2;
  c3 = Tan[α];

  c1 * Sqrt[(σλ * (λ * 10^-9) / (Sqrt[c2 - (λ * 10^-9) ^2])) ^2 + (c3 * σλ) ^2] * 10^9
 ](*error in linear dispersion relation of the monochromater in nm/px*)

σRavg[λ_] := Module[{dx, L, d, m, α, c, c1, c2, c3},
  dx = 12.8 / 10^6;
  L = 1;
  d = 1 / (2400 * 10^3);
  m = 1;
  α = 5.25 * π / 180;
  c = 3 * 10^8;

  c1 = dx / 2 * L;
  c2 = (2 * d * Cos[α] / m) ^2;
  c3 = Tan[α];

  c1 * Sqrt[(σλavg * (λ * 10^-9) / (Sqrt[c2 - (λ * 10^-9) ^2])) ^2 + (c3 * σλavg) ^2] * 10^9
 ](*error in linear dispersion relation of the monochromater in nm/px*)

Dop[λ_] := ((R[λ] / λ) * 3 * 10^8) / 1000 (*Dopler resolution in (km/s)/px*)

σDop[λ_] := Dop[λ] * Sqrt[(σR[λ] / R[λ]) ^2 + (σλ / λ) ^2]
```

Shot Settings

```
calwavelength = 656.3;(*spectrometer wavelength for calibration shot*)
shotwavelength = 656.3;(*spectrometer wavelength during shot*)
emissionwavelength = 656.3;(*emission line wavelength*)
avgwavelength = (calwavelength + shotwavelength) / 2;
correction = (shotwavelength - calwavelength) / R[avgwavelength]
  (*shift in pixels to correct for the calibration image having been taken at a slightly different spectrometer
   central wavelength than than the plasma shot*)

σcorrection =
 If[correction == 0, 0,
  Abs[correction] * Sqrt[(2 * (σλavg * 10^9) / (shotwavelength - calwavelength)) ^2 + (σRavg[avgwavelength] / R[avgwavelength]) ^2]]
```

Previously calculated user settings for various reference shots

```
Hαref = {-10, -10, 25, -9, 65, -8};(*settings for approxcenterfunction for Hα reference*)
ref7154 = {0, 0, 0, 0, 0, 0};(*settings for reference 7154*)
ref7183 = {0, 0, 0, 0, 0, 0};(*settingso for reference 7183*)
ref7225 = {-10, 0, -10, 0, -10, 0};(*settings for reference 7225*)
ref7242 = {-30, -7, 7, -7, 55, -7};(*settings for reference 7243*)
ref7243 = {-30, -7, 7, -7, 55, -7};(*settings for reference 7243*)

Hαrectangle = {80, 25};(*recangle dimensions width x height for Hα reference and subsequent shots*)
rectangle7154 = {100, 25};
rectangle7183 = {100, 25};
rectangle7225 = {100, 25};
rectangle7242 = {60, 25};
rectangle7243 = {60, 25};
```

Define binning grid

```
approxcenterleft = {{63, 951}, {64, 927}, {66, 901}, {68, 877}, {71, 851}, {73, 826}, {73, 800}, {73, 776}, {68, 748},
   {65, 725}, {63, 699}, {61, 671}, {62, 647}, {63, 619}, {63, 593}, {63, 568}, {68, 544}, {65, 520}, {67, 494},
   {72, 470}, {71, 445}, {74, 420}, {73, 395}, {66, 347}, {68, 322}, {69, 297}, {69, 271}, {69, 243}, {69, 218}, {69, 191}, {69, 166}};
approxcentercenter = {{487, 881}, {486, 855}, {487, 831}, {486, 805}, {486, 781}, {489, 755}, {489, 730}, {489, 705},
   {486, 679}, {487, 656}, {488, 623}, {487, 595}, {490, 565}, {488, 539}, {487, 514}, {490, 488}, {487, 453}, {488, 427},
   {490, 403}, {490, 378}, {492, 352}, {493, 328}, {493, 303}, {495, 277}, {481, 254}, {481, 229}, {483, 203}, {484, 177},
   {486, 153}, {487, 126}, {486, 99}, {481, 67}};
approxcenterright = {{871, 896}, {883, 851}, {883, 824}, {884, 800}, {886, 774}, {886, 749}, {886, 725}, {886, 701},
   {885, 676}, {885, 650}, {884, 625}, {883, 601}, {882, 579}, {882, 552}, {880, 528}, {878, 499}, {881, 476}, {884, 452},
   {881, 425}, {882, 401}, {881, 376}, {882, 350}, {881, 324}, {877, 275}, {878, 252}, {878, 227}, {878, 200}, {877, 175},
   {876, 150}, {876, 125}, {876, 100}};
 (*base background grid - DO NOT ADJUST*)

bundlelist = {{1, 2, 3, 4, 5, 6, 7, 8}, {9, 10, 11, 12, 13, 14, 15, 16}, {17, 18, 19, 20, 21, 22, 23}, {24, 25, 26, 27, 28, 29, 30, 31},
   {32, 33, 34, 35, 36, 37, 38, 39}, {40, 41, 42, 43, 44, 45, 46, 47}, {48, 49, 50, 51, 52, 53, 54, 55}, {56, 57, 58, 59, 60, 61, 62, 63},
   {64, 65, 66, 67, 68, 69, 70, 71}, {72, 73, 74, 75, 76, 77, 78, 79}, {80, 81, 82, 83, 84, 85, 86}, {87, 88, 89, 90, 91, 92, 93, 94}};
 (*this list will be used in later functions to assosciate a bundle and sight line with the approxcenterlist*)

approxcenter[{leftx_, lefty_, centerx_, centery_, rightx_, righty_}] :=
 Join[Table[{approxcenterleft[[i, 1]] + leftx, approxcenterleft[[i, 2]] - lefty}, {i, 1, Length[approxcenterleft]}],
  Table[{approxcentercenter[[i, 1]] + centerx, approxcentercenter[[i, 2]] - centery}, {i, 1, Length[approxcentercenter]}],
  Table[{approxcenterright[[i, 1]] + rightx, approxcenterright[[i, 2]] - righty}, {i, 1, Length[approxcenterright]}]]
  (*function allows you to adjust the columns up down left or right independently then combines them into a total list*)

rectangle[xcenter_, ycenter_] :=
 Graphics[{EdgeForm[{Thick, Red}], FaceForm[],
   Rectangle[{xcenter - Round[rectangledimensions[[1]] / 2], (1024 - ycenter) - Round[rectangledimensions[[2]] / 2]},
    {xcenter + Round[rectangledimensions[[1]] / 2], (1024 - ycenter) + Round[rectangledimensions[[2]] / 2]}]}]
  (*places a rectangle of the specified dimensions with a center at (xcenter, ycenter)*)
```

Select settings

```
approxcenterlist = approxcenter[Hαref];(*define the approx centerlist to be used - Specify reference settings to use*)
rectangledimensions = Hαrectangle;(*defines the size in pixels of each rectangular box*)
rectangletable = Table[rectangle[approxcenterlist[[i, 1]], approxcenterlist[[i, 2]]], {i, 1, Length[approxcenterlist]}];
(*creates a table of rectangles that defines the grid*)
```

```
Binplot[reffile_, shotfile_, bundle_, viewline_, wavelength_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, refline, σx0ref, shotbinnedlist, shotgaussian, shotcenter, shotline, σx0shot, shift, σshift, β},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  refline = Graphics[{Black, Thick, Line[{{refcenter, 0}, {refcenter, 150000}}]}];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]](**1.9847231860133867`*);

  shotbinnedlist = Table[{i, binnedvalue[shot, i]}, {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 30 > σ > 9,
     startcolumn + Round[correction] < x0 < endcolumn + Round[correction]},
    {{a, Max[shotbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[Length[shotbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  shotline = Graphics[{Black, Thick, Dashed, Line[{{shotcenter, 0}, {shotcenter, 150000}}]}];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]](**1.9847231860133867`*);

  shift = -1 * (shotcenter - refcenter) * Dop[wavelength];
  (*σshift = Sqrt[σx0ref^2+σx0shot^2]*Dop[wavelength];*)
  σshift =
   Abs[shift] * Sqrt[((σx0ref^2 + σx0shot^2 + σcorrection^2) / (shotcenter - refcenter)^2) + (σDop[wavelength] / Dop[wavelength])^2];

  β = shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 2]];
  (*standard noise in the signal with no intensity*)

 shotgaussian["ParameterConfidenceIntervalTable"] ×
 refgaussian["ParameterConfidenceIntervalTable"] ×
 Labeled[Show[ListPlot[{refbinnedlist, shotbinnedlist}, PlotRange → {10000, 150000}, PlotStyle → Black,
    PlotTheme -> "Monochrome", GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True],
   Plot[{refgaussian[x], shotgaussian[x]}, {x, startcolumn + correction, endcolumn + correction}, PlotTheme -> "Monochrome",
    GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True, PlotRange → All], shotline, refline, ImageSize → 250],
  PlusMinus[shift, σshift] "km/s", Top
 ]


Binplottable[reffile_, shotfile_, bundle_, wavelength_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[Binplot[reffile, shotfile, bundle, i, wavelength, correction], {i, 1, l}]
 ]
```

```
waveplot[reffile_, shotfile_, bundle_, viewline_, λe_, λm_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, refσ, refσσ, refline, newrefbinnedlist, shotbinnedlist, shotgaussian, shotcenter, shotamp, shotσ,
   shotσσ, shotb, shotline, instrum, m, c, k, T, σT, Tev, σTev},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  refσ = refgaussian["ParameterTableEntries"][[3, 1]];
  refσσ = refgaussian["ParameterTableEntries"][[3, 2]];
  refline = Graphics[{Black, Thick, Line[{{λe, 0}, {λe, 100000}}]}];

  newrefbinnedlist = Table[{(refcenter - i) * R[λm] + λe, binnedvalue[ref, i]},
    {i, startcolumn + Round[correction], endcolumn + Round[correction]}];


  shotbinnedlist = Table[{(refcenter - i) * R[λm] + λe, binnedvalue[shot, i]},
    {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, R[λm] * 30 > σ > R[λm] * 4, λe - 0.15 < x0 < λe + 0.15},
    {{a, Max[shotbinnedlist]}, {x0, 723.6}, {σ, R[λm] * 20}, {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[-1, 2]]}]}},
    x, MaxIterations → 5000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  shotamp = shotgaussian["ParameterTableEntries"][[1, 1]];
  shotσ = shotgaussian["ParameterTableEntries"][[3, 1]];
  shotσσ = shotgaussian["ParameterTableEntries"][[3, 2]];
  shotb = shotgaussian["ParameterTableEntries"][[4, 1]];
  shotline = Graphics[{Black, Thick, Dashed, Line[{{shotcenter, 0}, {shotcenter, 100000}}]}];
 instrum = shotamp * (refσ * R[λm]) / shotσ * Evaluate[PDF[NormalDistribution[shotcenter, refσ * R[λm]], x]] + shotb;

 m = 1.67 * 10^-27;
 c = 2.998 * 10^8;
 k = 1.380648 * 10^-23;

 T = m * c^2 / (k * (λe * 10^-9)^2) * ((shotσ * 10^-9)^2 - (refσ * R[λm] * 10^-9)^2);
 (*σT=m*c^2/(k*(λe*10^-9)^2)*
    (Sqrt[(2*(shotσσ*10^-9)/(shotσ*10^-9)*(shotσ*10^-9)^2)^2+
       (2*(refσσ*R[λm]*10^-9)/(refσ*R[λm]*10^-9)*(refσ*R[λm]*10^-9)^2)^2])*)
 σT =
  T *
   Sqrt[(2 * σR[λm] / R[λm])^2 + 4 (((shotσσ * 10^-9) (shotσ * 10^-9))^2 + ((refσσ * R[λm] * 10^-9) (refσ * R[λm] * 10^-9))^2) /
       ((shotσ * 10^-9)^2 - (refσ * R[λm] * 10^-9)^2)^2];


 Tev = T * 8.621738 * 10^-5;
 σTev = σT * 8.621738 * 10^-5;

 shotgaussian["ParameterConfidenceIntervalTable"]


  ×
  Labeled[Show[ListPlot[{shotbinnedlist, newrefbinnedlist}, PlotRange → {10000, 40000}, PlotStyle → Black,
     PlotTheme -> "Monochrome", GridLines → Automatic, PlotMarkers → {{△, 15}, {■, 15}}, BaseStyle → {FontSize → 25}, Frame → True],
    Plot[{refgaussian[refcenter - (x - λe) / R[λm]], shotgaussian[x], instrum},
     {x, (refcenter - (startcolumn + Round[correction])) * R[λm] + λe, (refcenter - (endcolumn + Round[correction])) * R[λm] + λe},
     PlotTheme -> "Monochrome", GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True], shotline, refline, ImageSize → 500],
   PlusMinus[Tev, σTev], Top]
 (*Show[Plot[{refgaussian[refcenter-(x-λe)/R[λm]]},
   {x,(refcenter-(startcolumn+Round[correction]))*R[λm]+λe,(refcenter-(endcolumn+Round[correction]))*R[λm]+λe},
   PlotStyle→Dashed,PlotTheme->"Monochrome",GridLines→Automatic,BaseStyle→{FontSize→25},Frame→True]]*)
](*similar to bin plot but plots the binned intensity in wavelength space*)
```

```
waveplottable[reffile_, shotfile_, bundle_, λe_, λm_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[waveplot[reffile, shotfile, bundle, i, λe, λm, correction], {i, 1, l}]
 ]


lineintegrateddata[reffile_, shotfile_, bundle_, viewline_, λe_, λm_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, refσ, σx0ref, refσσ, shotbinnedlist, shotgaussian, shotcenter, σx0shot, shotσ, shotσσ, shift, σshift,
   m, c, k, T, σT, Tev, σTev},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] (**1.9847231860133867`*);
  refσ = refgaussian["ParameterTableEntries"][[3, 1]];
  refσσ = refgaussian["ParameterTableEntries"][[3, 2]];



  shotbinnedlist = Table[{i, binnedvalue[shot, i]}, {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 30 > σ > 9,
     startcolumn + Round[correction] < x0 < endcolumn + Round[correction]},
    {{a, Max[shotbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[Length[shotbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] (**1.9847231860133867`*);
  shotσ = shotgaussian["ParameterTableEntries"][[3, 1]];
  shotσσ = shotgaussian["ParameterTableEntries"][[3, 2]];


  shift = -1 * (shotcenter - refcenter) * Dop[λm];
  (*σshift = Sqrt[σx0ref^2+σx0shot^2]*Dop[λm];*)
  σshift = Abs[shift] * Sqrt[((σx0ref^2 + σx0shot^2 + σcorrection^2) / (shotcenter - refcenter)^2) + (σDop[λm] / Dop[λm])^2];

  m = 1.67 * 10^-27;
  c = 2.998 * 10^8;
  k = 1.380648 * 10^-23;

  T = m * c^2 / (k * (λe * 10^-9)^2) * ((shotσ * R[λm] * 10^-9)^2 - (refσ * R[λm] * 10^-9)^2);
  σT = T * Sqrt[(2 * σR[λm] / R[λm])^2 + 4 (((shotσσ) (shotσ))^2 + ((refσσ) (refσ))^2) / ((shotσ)^2 - (refσ)^2)^2];

  Tev = T * 8.621738 * 10^-5;
  σTev = σT * 8.621738 * 10^-5;

  {shift, σshift, Tev, σTev}
 ]
```

```
lineinttable[reffile_, shotfile_, bundle_, λe_, λm_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[lineintegrateddata[reffile, shotfile, bundle, i, λe, λm, correction], {i, 1, l}]
 ]
```

Shot Data (shift [km/s], σshift, Tev, σTev)

Frame1

```
b1data1 = Quiet[lineinttable[ref, frame1, 1, emissionwavelength, shotwavelength, correction]];(*data*)
b2data1 = Quiet[lineinttable[ref, frame1, 2, emissionwavelength, shotwavelength, correction]];
b3data1 = Quiet[lineinttable[ref, frame1, 3, emissionwavelength, shotwavelength, correction]];
b4data1 = Quiet[lineinttable[ref, frame1, 4, emissionwavelength, shotwavelength, correction]];
b5data1 = Quiet[lineinttable[ref, frame1, 5, emissionwavelength, shotwavelength, correction]];
b6data1 = Quiet[lineinttable[ref, frame1, 6, emissionwavelength, shotwavelength, correction]];
b7data1 = Quiet[lineinttable[ref, frame1, 7, emissionwavelength, shotwavelength, correction]];
b8data1 = Quiet[lineinttable[ref, frame1, 8, emissionwavelength, shotwavelength, correction]];
b9data1 = Quiet[lineinttable[ref, frame1, 9, emissionwavelength, shotwavelength, correction]];
b10data1 = Quiet[lineinttable[ref, frame1, 10, emissionwavelength, shotwavelength, correction]];
b11data1 = Quiet[lineinttable[ref, frame1, 11, emissionwavelength, shotwavelength, correction]];
b12data1 = Quiet[lineinttable[ref, frame1, 12, emissionwavelength, shotwavelength, correction]];
```

Frame2

```
b1data2 = Quiet[lineinttable[ref, frame2, 1, emissionwavelength, shotwavelength, correction]];
b2data2 = Quiet[lineinttable[ref, frame2, 2, emissionwavelength, shotwavelength, correction]];
b3data2 = Quiet[lineinttable[ref, frame2, 3, emissionwavelength, shotwavelength, correction]];
b4data2 = Quiet[lineinttable[ref, frame2, 4, emissionwavelength, shotwavelength, correction]];
b5data2 = Quiet[lineinttable[ref, frame2, 5, emissionwavelength, shotwavelength, correction]];
b6data2 = Quiet[lineinttable[ref, frame2, 6, emissionwavelength, shotwavelength, correction]];
b7data2 = Quiet[lineinttable[ref, frame2, 7, emissionwavelength, shotwavelength, correction]];
b8data2 = Quiet[lineinttable[ref, frame2, 8, emissionwavelength, shotwavelength, correction]];
b9data2 = Quiet[lineinttable[ref, frame2, 9, emissionwavelength, shotwavelength, correction]];
b10data2 = Quiet[lineinttable[ref, frame2, 10, emissionwavelength, shotwavelength, correction]];
b11data2 = Quiet[lineinttable[ref, frame2, 11, emissionwavelength, shotwavelength, correction]];
b12data2 = Quiet[lineinttable[ref, frame2, 12, emissionwavelength, shotwavelength, correction]];
```

The following notebook was used for the analysis of shot #7150, any identical functions repeated from the previous notebook are omitted. Some functions may have the same name but differ subtly from the previous notebook.

Spectroscopic Lines of sight

```
centerzero = {-0.259, 0.433, 0};
centerend = {-0.259, -0.615, 0};

b1f = {0.01, 0.531, 0.294};(*bundle1 footpoint*)
b1l1 = {-0.354, -0.607, -0.149};
b1l2 = {-0.4, -0.607, -0.161};
b1l3 = {-0.455, -0.607, -0.176};
b1l4 = {-0.504, -0.607, -0.189};
b1l5 = {-0.547, -0.607, -0.201};
b1l6 = {-0.611, -0.607, -0.207};
b1l7 = {-0.659, -0.607, -0.221};
b1l8 = {-0.685, -0.607, -0.239};
bundle1 = {b1f, b1l1, b1l2, b1l3, b1l4, b1l5, b1l6, b1l7, b1l8};

b2f = {0.223, 0.370, 0.281};
b2l1 = {-0.543, -0.611, -0.215};
b2l2 = {-0.603, -0.617, -0.219};
b2l3 = {-0.685, -0.607, -0.239};
b2l4 = {-0.708, -0.595, -0.264};
b2l5 = {-0.735, -0.553, -0.271};
b2l6 = {-0.763, -0.525, -0.262};
b2l7 = {-0.789, -0.489, -0.265};
b2l8 = {-0.812, -0.450, -0.270};
bundle2 = {b2f, b2l1, b2l2, b2l3, b2l4, b2l5, b2l6, b2l7, b2l8};

b3f = {0.371, 0.150, 0.282};
b3l1 = {-0.783, -0.466, -0.305};
b3l2 = {-0.810, -0.424, -0.302};
b3l3 = {-0.823, -0.382, -0.318};
b3l4 = {-0.841, -0.339, -0.318};
b3l5 = {-0.852, -0.297, -0.324};
b3l6 = {-0.860, -0.247, -0.331};
b3l7 = {-0.867, -0.202, -0.333};
bundle3 = {b3f, b3l1, b3l2, b3l3, b3l4, b3l5, b3l6, b3l7};

b4f = {0.423, -0.110, 0.283};
b4l1 = {-0.866, -0.264, -0.313};
b4l2 = {-0.879, -0.218, -0.306};
b4l3 = {-0.881, -0.180, -0.312};
b4l4 = {-0.893, -0.141, -0.291};
b4l5 = {-0.892, -0.080, -0.291};
b4l6 = {-0.876, -0.011, -0.309};
b4l7 = {-0.879, 0.030, -0.286};
b4l8 = {-0.868, 0.073, -0.282};
bundle4 = {b4f, b4l1, b4l2, b4l3, b4l4, b4l5, b4l6, b4l7, b4l8};

b5f = {0.373, -0.371, 0.285};
b5l1 = {-0.883, -0.047, -0.304};
b5l2 = {-0.873, 0.014, -0.304};
b5l3 = {-0.862, 0.062, -0.302};
b5l4 = {-0.846, 0.112, -0.299};
b5l5 = {-0.828, 0.156, -0.295};
b5l6 = {-0.799, 0.204, -0.302};
b5l7 = {-0.783, 0.241, -0.289};
b5l8 = {-0.760, 0.278, -0.280};
bundle5 = {b5f, b5l1, b5l2, b5l3, b5l4, b5l5, b5l6, b5l7, b5l8};

b6f = {0.225, -0.593, 0.285};
b6l1 = {-0.788, 0.250, -0.266};
b6l2 = {-0.766, 0.285, -0.257};
b6l3 = {-0.743, 0.320, -0.245};
b6l4 = {-0.719, 0.363, -0.237};
b6l5 = {-0.675, 0.392, -0.245};
b6l6 = {-0.617, 0.392, -0.220};
b6l7 = {-0.546, 0.392, -0.190};
b6l8 = {-0.473, 0.392, -0.159};
bundle6 = {b6f, b6l1, b6l2, b6l3, b6l4, b6l5, b6l6, b6l7, b6l8};
```

```
b7f = {0.024, -0.110, 0.681};
b7l1 = {-0.398, -0.110, -0.721};
b7l2 = {-0.419, -0.096, -0.680};
b7l3 = {-0.461, -0.093, -0.669};
b7l4 = {-0.495, -0.101, -0.658};
b7l5 = {-0.534, -0.083, -0.642};
b7l6 = {-0.567, -0.098, -0.627};
b7l7 = {-0.600, -0.094, -0.609};
b7l8 = {-0.642, -0.090, -0.583};
bundle7 = {b7f, b7l1, b7l2, b7l3, b7l4, b7l5, b7l6, b7l7, b7l8};

b8f = {0.263, -0.110, 0.521};
b8l1 = {-0.641, -0.106, -0.585};
b8l2 = {-0.674, -0.103, -0.562};
b8l3 = {-0.706, -0.102, -0.537};
b8l4 = {-0.729, -0.087, -0.515};
b8l5 = {-0.756, -0.080, -0.489};
b8l6 = {-0.779, -0.088, -0.465};
b8l7 = {-0.806, -0.103, -0.434};
b8l8 = {-0.829, -0.095, -0.402};
bundle8 = {b8f, b8l1, b8l2, b8l3, b8l4, b8l5, b8l6, b8l7, b8l8};

b9f = {0.478, -0.110, 0.169};
b9l1 = {-0.893, -0.170, -0.287};
b9l2 = {-0.911, -0.143, -0.247};
b9l3 = {-0.920, -0.165, -0.218};
b9l4 = {-0.933, -0.164, -0.177};
b9l5 = {-0.942, -0.164, -0.135};
b9l6 = {-0.949, -0.155, -0.097};
b9l7 = {-0.953, -0.162, -0.069};
b9l8 = {-0.964, -0.173, -0.010};
bundle9 = {b9f, b9l1, b9l2, b9l3, b9l4, b9l5, b9l6, b9l7, b9l8};

b10f = {0.424, -0.110, -0.281};
b10l1 = {-0.946, -0.077, 0.116};
b10l2 = {-0.933, -0.068, 0.175};
b10l3 = {-0.919, -0.092, 0.226};
b10l4 = {-0.907, -0.076, 0.255};
b10l5 = {-0.890, -0.101, 0.298};
b10l6 = {-0.871, -0.083, 0.333};
b10l7 = {-0.852, -0.101, 0.368};
b10l8 = {-0.832, -0.117, 0.399};
bundle10 = {b10f, b10l1, b10l2, b10l3, b10l4, b10l5, b10l6, b10l7, b10l8};

b11f = {0.264, -0.110, -0.520};
b11l1 = {-0.843, -0.102, 0.382};
b11l2 = {-0.817, -0.094, 0.419};
b11l3 = {-0.793, -0.081, 0.448};
b11l4 = {-0.767, -0.075, 0.477};
b11l5 = {-0.735, -0.075, 0.509};
b11l6 = {-0.704, -0.088, 0.537};
b11l7 = {-0.677, -0.073, 0.558};
bundle11 = {b11f, b11l1, b11l2, b11l3, b11l4, b11l5, b11l6, b11l7};

b12f = {0.026, -0.110, -0.680};
b12l1 = {-0.666, -0.021, 0.560};
b12l2 = {-0.626, -0.009, 0.584};
b12l3 = {-0.599, -0.009, 0.601};
b12l4 = {-0.562, -0.009, 0.620};
b12l5 = {-0.524, -0.007, 0.637};
b12l6 = {-0.485, -0.008, 0.652};
b12l7 = {-0.442, -0.006, 0.665};
b12l8 = {-0.403, -0.008, 0.675};
bundle12 = {b12f, b12l1, b12l2, b12l3, b12l4, b12l5, b12l6, b12l7, b12l8};

bundles = {bundle1, bundle2, bundle3, bundle4, bundle5, bundle6, bundle7, bundle8, bundle9, bundle10, bundle11, bundle12};
```

Useful functions for later use

```
norm[{x_, y_, z_}] = Sqrt[x^2 + y^2 + z^2];

unitvec[{x1_, y1_, z1_}, {x2_, y2_, z2_}] = ({x2, y2, z2} - {x1, y1, z1}) / norm[{x2, y2, z2} - {x1, y1, z1}];

pointlinedistance[lp1_, lp2_, p_] = Norm[Cross[lp1 - lp2, lp1 - p]] / Norm[lp2 - lp1];

minimizer[lp1_, lp2_, px_, pz_] := FindArgMin[pointlinedistance[lp1, lp2, {px, py, pz}], py, AccuracyGoal → 4][[1]]


lineintegrateddata[reffile_, shotfile_, bundle_, viewline_, wavelength_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, σx0ref, shotbinnedlist, shotgaussian, shotcenter, σx0shot, bshot, shift, σshift, β, backgroundintensity,
   σbackgroundintensity, area, σarea, σbackgroundremovedarea},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + m * x + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {m, (refbinnedlist[[1, 2]] - refbinnedlist[[-1, 2]]) / (endcolumn - startcolumn)},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] (**1.9847231860133867`*);

  shotbinnedlist = Table[{i, binnedvalue[shot, i]}, {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 30 > σ > 9,
     startcolumn + Round[correction] < x0 < endcolumn + Round[correction]},
    {{a, Max[shotbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[Length[shotbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] (**1.9847231860133867`*);
  bshot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 1]];

  shift = -1 * (shotcenter - refcenter) * Dop[wavelength];
  (*σshift = Sqrt[σx0ref^2+σx0shot^2]*Dop[wavelength];*)
  σshift =
   Abs[shift] * Sqrt[((σx0ref^2 + σx0shot^2 + σcorrection^2) / (shotcenter - refcenter)^2) + (σDop[wavelength] / Dop[wavelength])^2];

  β = shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 2]];

  backgroundintensity = bshot * (75 * 2);
  (*integrate the background to find contribution to the total area under the curve*)
  σbackgroundintensity = β * (75 * 2);

  area = NIntegrate[shotgaussian[x], {x, shotcenter - 75, shotcenter + 75}];
  σarea = (NIntegrate[β (1 + Sqrt[(shotgaussian[x] - bshot) / bshot]), {x, shotcenter - 75, shotcenter + 75}]);

  σbackgroundremovedarea = Sqrt[σarea^2 + σbackgroundintensity^2];



  {shift, σshift, area, σarea, area - backgroundintensity, (shotgaussian[shotcenter] - bshot) / (β), σbackgroundremovedarea}
 ]
```

```
lineinttable[reffile_, shotfile_, bundle_, wavelength_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[lineintegrateddata[reffile, shotfile, bundle, i, wavelength, correction], {i, 1, l}]
 ]


Binplot[reffile_, shotfile_, bundle_, viewline_, wavelength_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, refline, σx0ref, shotbinnedlist, shotgaussian, shotcenter, shotline, σx0shot, shift, σshift, β},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + m * x + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {m, (refbinnedlist[[1, 2]] - refbinnedlist[[-1, 2]]) / (endcolumn - startcolumn)},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  refline = Graphics[{Black, Thick, Line[{{refcenter, 0}, {refcenter, 100000}}]}];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867`;

  shotbinnedlist = Table[{i, binnedvalue[shot, i]}, {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, 30 > σ > 9,
     startcolumn + Round[correction] < x0 < endcolumn + Round[correction]},
    {{a, Max[shotbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[Length[shotbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  shotline = Graphics[{Black, Thick, Dashed, Line[{{shotcenter, 0}, {shotcenter, 100000}}]}];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867`;

  shift = -1 * (shotcenter - refcenter) * Dop[wavelength];
  (*σshift = Sqrt[σx0ref^2+σx0shot^2]*Dop[wavelength];*)
  σshift =
   Abs[shift] * Sqrt[((σx0ref^2 + σx0shot^2 + σcorrection^2) / (shotcenter - refcenter)^2) + (σDop[wavelength] / Dop[wavelength])^2];

  β = shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 2]];
  (*standard noise in the signal with no intensity*)
 shotgaussian["ParameterConfidenceIntervalTable"] ×
  refgaussian["ParameterConfidenceIntervalTable"] ×
  Labeled[Show[ListPlot[{refbinnedlist, shotbinnedlist}, PlotRange → {10000, 60000}, PlotStyle → Black,
     PlotTheme -> "Monochrome", GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True],
    Plot[{refgaussian[x], shotgaussian[x],
      shotgaussian[x] +
       β (1 + Sqrt[(shotgaussian[x] - shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 1]]) /
            shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 1]]]),
      shotgaussian[x] -
       β (1 + Sqrt[(shotgaussian[x] - shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 1]]) /
            shotgaussian["ParameterConfidenceIntervalTableEntries"][[4, 1]]])},
     {x, startcolumn + correction, endcolumn + correction}, PlotTheme -> "Monochrome", GridLines → Automatic,
     BaseStyle → {FontSize → 25}, Frame → True, PlotRange → All], shotline, refline, ImageSize → 250],
   PlusMinus[shift, σshift] "km/s", Top
  ]
```

```
Binplottable[reffile_, shotfile_, bundle_, wavelength_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[Binplot[reffile, shotfile, bundle, i, wavelength, correction], {i, 1, l}]
 ]



waveplot[reffile_, shotfile_, bundle_, viewline_, λe_, λm_, correction_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, refline, shotbinnedlist, shotgaussian, shotcenter, shotline},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + m * x + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {m, (refbinnedlist[[1, 2]] - refbinnedlist[[-1, 2]]) / (endcolumn - startcolumn)},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  refline = Graphics[{Black, Thick, Line[{{λe, 0}, {λe, 100000}}]}];

  shotbinnedlist = Table[{(refcenter - i) * R[λm] + λe, binnedvalue[shot, i]},
    {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, R[λm] * 30 > σ > 0, λe - 0.15 < x0 < λe + 0.15},
    {{a, Max[shotbinnedlist]}, {x0, 723.6}, {σ, R[λm] * 10}, {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[-1, 2]]}]}},
    x, MaxIterations → 5000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  shotline = Graphics[{Black, Thick, Dashed, Line[{{shotcenter, 0}, {shotcenter, 100000}}]}];
  shotgaussian["ParameterConfidenceIntervalTable"]
   Show[ListPlot[shotbinnedlist, PlotRange → {10000, 20000}, PlotStyle → Black, PlotTheme -> "Monochrome",
     GridLines → Automatic, PlotMarkers → {{△, 15}}, BaseStyle → {FontSize → 25}, Frame → True],
    Plot[shotgaussian[x], {x, (refcenter - (startcolumn + Round[correction])) * R[λm] + λe,
      (refcenter - (endcolumn + Round[correction])) * R[λm] + λe}, PlotStyle → Dashed, PlotTheme -> "Monochrome",
     GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True], shotline, refline, ImageSize → 250]
 ](*similar to bin plot but plots the binned intensity in wavelength space*)

waveplottable[reffile_, shotfile_, bundle_, λe_, λm_, correction_] := Module[{l},
  l = Length[bundlelist[[bundle]]];
  Table[waveplot[reffile, shotfile, bundle, i, λe, λm, correction], {i, 1, l}]
 ]
```

```
threedplot[reffile_, shotfile_, bundle_, viewline_, λe_, λm_, correction_, level_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, σx0ref, shotbinnedlist, shotgaussian, shotcenter, σx0shot, σshift, shift, finallist},
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + m * x + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {m, (refbinnedlist[[1, 2]] - refbinnedlist[[-1, 2]]) / (endcolumn - startcolumn)},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867` * R[λm];


  shotbinnedlist = Table[{(refcenter - i) * R[λm] + λe, binnedvalue[shot, i]},
    {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, R[λm] * 30 > σ > 0, λe - 0.15 < x0 < λe + 0.15},
    {{a, Max[shotbinnedlist]}, {x0, 723.6}, {σ, R[λm] * 10}, {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[-1, 2]]}]}},
    x, MaxIterations → 5000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867`;

  σshift = Sqrt[σx0ref^2 + σx0shot^2] / λe * 3 * 10^8 / 1000;
  shift = PlusMinus[(shotcenter - λe) / λe * 3 * 10^8 / 1000, σshift];

  finallist = Table[{shotbinnedlist[[i, 1]], level, shotbinnedlist[[i, 2]]}, {i, 1, Length[shotbinnedlist]}];

 {Show[ListPointPlot3D[finallist, PlotRange → {{λe - 0.3, λe + 0.4}, {0, 9}, {14000, 30000}}, PlotTheme → "Detailed",
    PlotStyle → {Black, PointSize → Large}, BaseStyle → {FontSize → 50}, AxesStyle → Thickness[0.005],
    FaceGrids → {{{-1, 0, 0}, {{1, 2, 3, 4, 5, 6, 7, 8}, {15000, 20000, 25000, 30000}}},
      {{0, 0, -1}, {{723.4, 723.6, 723.8, 724}, {1, 2, 3, 4, 5, 6, 7, 8}}},
      {{0, 1, 0}, {{723.4, 723.6, 723.8, 724}, {15000, 20000, 25000, 30000}}}},
    FaceGridsStyle → Directive[Thick, Dashed]],
   ParametricPlot3D[{x, level, shotgaussian[x]}, {x, (refcenter - (startcolumn + Round[correction])) * R[λm] + λe,
     (refcenter - (endcolumn + Round[correction])) * R[λm] + λe}, PlotStyle → Directive[Black, Thickness[0.005]]], ImageSize → 800],
  shotcenter, shift, Sqrt[σx0ref^2 + σx0shot^2]}

](*Plots the binned intensity as a function of wavelength onto a 3D grid where the level specifies the view chord number*)
```

```
chordplot[reffile_, shotfile_, bundle_, λe_, λm_, correction_, baselevel_] :=
 Module[{l, shotcenters, shifts, shifttext, errorbars, refline, centerline, extragrids1, extragrids2, points},
  l = Length[bundlelist[[bundle]]];
  shotcenters = Table[threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]], {i, 1, l}];
  shifts = Table[threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[3]], {i, 1, l}];
  shifttext = Table[Graphics3D[Text[NumberForm[shifts[[i]], {2, 1}], {λe + 0.25, i, baselevel}, BaseStyle → {Black, FontSize → 45}]],
    {i, 1, l}];
  errorbars = Table[
    {Graphics3D[
      {Red, Thickness[0.009],
       Line[{{threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] -
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i, baselevel},
          {threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] +
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i, baselevel}}]}],
     Graphics3D[
      {Red, Thickness[0.005],
       Line[{{threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] -
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i - .1, baselevel},
          {threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] -
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i + .1, baselevel}}]}],
     Graphics3D[
      {Red, Thickness[0.005],
       Line[{{threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] +
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i - .1, baselevel},
          {threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[2]] +
            threedplot[reffile, shotfile, bundle, i, λe, λm, correction, 1][[4]], i + .1, baselevel}}]}]}, {i, 1, l}];
  refline = Graphics3D[{Thickness[0.01], Red, Dashed, Line[{{λe, 0, baselevel}, {λe, 9, baselevel}}]}];
  centerline = Graphics3D[{Red, Thickness[0.008], Line[Table[{shotcenters[[i]], i, baselevel}, {i, 1, l}]]}];
  (*points = Graphics3D[{PointSize[0.5],Point[Table[{shotcenters[[i]],i,0},{i,1,l}]]}];*)
  Labeled[Show[Table[threedplot[reffile, shotfile, bundle, i, λe, λm, correction, i][[1]], {i, 1, l}], errorbars,
    refline, centerline, (*points,*)shifttext, ImageSize → 1000], "Bundle" <> ToString[bundle], Top, LabelStyle → {FontSize → 30}]
 ](*generates a 3D grid of the binned intensity as a function of wavelength for all fibers in a bundle*)
```

```
stack[reffile_, shotfile_, bundle_, viewline_, λe_, λm_, correction_, level_] :=
 Module[{ref, shot, startrow, endrow, startcolumn, endcolumn, rowindex, binnedvalue, refbinnedlist, refgaussian,
   refcenter, σx0ref, shotbinnedlist, shotgaussian, shotcenter, σx0shot, max1, min, difference, finallist, max2, σshift, shift,}
  ref = reffile;
  shot = shotfile;
  startrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] - Round[rectangledimensions[[2]] / 2];
  endrow = approxcenterlist[[bundlelist[[bundle, viewline]], 2]] + Round[rectangledimensions[[2]] / 2];
  startcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] - Round[rectangledimensions[[1]] / 2];
  endcolumn = approxcenterlist[[bundlelist[[bundle, viewline]], 1]] + Round[rectangledimensions[[1]] / 2];
  rowindex[row_] = 2 + 1024 * row;
  binnedvalue[file_, column_] := Sum[file[[rowindex[i] + column]][[3]], {i, startrow, endrow}];

  refbinnedlist = Table[{i + correction, binnedvalue[ref, i]}, {i, startcolumn, endcolumn}];
  (*generates table of pixel position and binned intensity*)
  refgaussian = NonlinearModelFit[refbinnedlist, {a* Evaluate[PDF[NormalDistribution[x0, σ], x]] + m*x + b, a > 0, 20 > σ > 5},
    {{a, Max[refbinnedlist]}, {x0, 0.5 (startcolumn + endcolumn) + correction}, {σ, 10},
     {m, (refbinnedlist[[1, 2]] - refbinnedlist[[-1, 2]]) / (endcolumn - startcolumn)},
     {b, Mean[{refbinnedlist[[1, 2]], refbinnedlist[[Length[refbinnedlist], 2]]}]}}, x, MaxIterations → 1000];
  refcenter = refgaussian["ParameterTableEntries"][[2, 1]];
  σx0ref = refgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867` * R[λm];

  shotbinnedlist = Table[{(refcenter - i) * R[λm] + λe, binnedvalue[shot, i]},
    {i, startcolumn + Round[correction], endcolumn + Round[correction]}];
  shotgaussian = NonlinearModelFit[shotbinnedlist,
    {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, R[λm] * 30 > σ > 0, λe - 0.15 < x0 < λe + 0.15},
    {{a, Max[shotbinnedlist]}, {x0, 723.6}, {σ, R[λm] * 10}, {b, Mean[{shotbinnedlist[[1, 2]], shotbinnedlist[[-1, 2]]}]}},
    x, MaxIterations → 5000];
  shotcenter = shotgaussian["ParameterTableEntries"][[2, 1]];
  σx0shot = shotgaussian["ParameterConfidenceIntervalTableEntries"][[2, 2]] * 1.9847231860133867`;


  max1 = Max[shotbinnedlist];
  min = shotgaussian[(refcenter - (endcolumn + Round[correction])) * R[λm] + λe] / max1;
  difference = min - level;
  finallist = Table[{shotbinnedlist[[i, 1]], shotbinnedlist[[i, 2]] / max1 - difference}, {i, 1, Length[shotbinnedlist]}];

  max2 = FindMaximum[shotgaussian[x], {x, 723.6}][[1]];

  σshift = Sqrt[σx0ref^2 + σx0shot^2] / λe * 3 * 10^8 / 1000;
  shift = PlusMinus[(shotcenter - λe) / λe * 3 * 10^8 / 1000, σshift];

 {Show[ListPlot[finallist, PlotRange → {{λe - 0.2, λe + 0.3}, {0, 5}}, PlotTheme → "Scientific",
    PlotStyle → {Black, PointSize → Large}, BaseStyle → Directive[{FontSize → 50}], AxesStyle → Thickness[0.005],
    FrameTicks → {{None, None}, {Automatic, Automatic}}, FrameStyle → Directive[Thick]],
   Plot[shotgaussian[x] / max1 - difference, {x, (refcenter - (startcolumn + Round[correction])) * R[λm] + λe,
     (refcenter - (endcolumn + Round[correction])) * R[λm] + λe}, PlotStyle → Directive[Black, Thickness[0.005]]], ImageSize → 800],
  shotcenter, shift, Sqrt[σx0ref^2 + σx0shot^2]}
]
```

```
stacktable[reffile_, shotfile_, bundle_, λe_, λm_, correction_] :=
  Module[{l, offsetlist, smallshift, shotcenters, shifts, errors, shifttext, refline, centerline, minline, maxline, region},
   l = Length[bundlelist[[bundle]]];
   offsetlist = Subdivide[0.2, 4, 7];
   smallshift = 0;
   shotcenters = Table[stack[reffile, shotfile, bundle, i, emissionwavelength, shotwavelength, correction, offsetlist[[i]]][[2]],
     {i, 1, l}];
   shifts = Table[stack[reffile, shotfile, bundle, i, emissionwavelength, shotwavelength, correction, offsetlist[[i]]][[3]],
     {i, 1, l}];
   errors = Table[stack[reffile, shotfile, bundle, i, emissionwavelength, shotwavelength, correction, offsetlist[[i]]][[4]],
     {i, 1, l}];
   shifttext =
    Table[Graphics[Text[NumberForm[shifts[[i]], {2, 1}], {λe + 0.22, offsetlist[[i]]}, BaseStyle → {Black, FontSize → 45}]],
     {i, 1, l}];
   refline = Graphics[{Thickness[0.01], Red, Dashed, Line[{{λe, 0}, {λe, 10}}]}];
   centerline = Graphics[{Red, Thickness[0.008], Line[Table[{shotcenters[[i]], offsetlist[[i]] + smallshift}, {i, 1, l}]]}];
   minline =
    Graphics[{Red, Thickness[0.004], Line[Table[{shotcenters[[i]] - errors[[i]], offsetlist[[i]] + smallshift}, {i, 1, l}]]}];
   maxline =
    Graphics[{Red, Thickness[0.004], Line[Table[{shotcenters[[i]] + errors[[i]], offsetlist[[i]] + smallshift}, {i, 1, l}]]}];

   region =
    Graphics[
     {Red, Opacity[0.3], Polygon[Join[Table[{shotcenters[[i]] - errors[[i]], offsetlist[[i]] + smallshift}, {i, 1, l}],
        Reverse[Table[{shotcenters[[i]] + errors[[i]], offsetlist[[i]] + smallshift}, {i, 1, l}]]]]}];

   Show[Table[stack[reffile, shotfile, bundle, i, emissionwavelength, shotwavelength, correction, offsetlist[[i]]][[1]],
     {i, 1, l}], shifttext, refline, centerline, region]

  ];
```

```
areafit[list_] := Module[{max, gaussian, background, curvemax, xmax, sides, p1, p2, fwqm},
  max = Max[list];
  gaussian = NonlinearModelFit[list, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, b > 0},
    {{a, max}, {x0, 4}, {σ, 3}, {b, 1 * 10^6}}, x, MaxIterations → 5000];

  background = gaussian["ParameterTableEntries"][[4, 1]];

  curvemax = Quiet[FindMaximum[gaussian[x], x][[1]]];
  xmax = x /. Quiet[FindMaximum[gaussian[x], x][[2]]];

  sides = Quiet@Solve[gaussian[x] == (curvemax - background) / 4 + background];
  p1 = Re[x /. sides[[1]]];
  p2 = Re[x /. sides[[2]]];
  fwqm = Abs[p2 - p1];
  {Show[Plot[gaussian[x], {x, 0, 8}, PlotStyle → Black, PlotTheme -> "Monochrome", GridLines → Automatic,
     BaseStyle → {FontSize → 25}, Frame → True],
    ListPlot[{{p1, (curvemax - background) / 4 + background}, {p2, (curvemax - background) / 4 + background}},
     PlotStyle → Red, Joined → True]], xmax, fwqm}
 ]

areafit2[list_] := Module[{max, gaussian, background, curvemax, xmax, sides, p1, p2, fwhm},
  max = Max[list];
  gaussian = NonlinearModelFit[list, {a * Evaluate[PDF[NormalDistribution[x0, σ], x]] + b, a > 0, b > 0},
    {{a, max}, {x0, 4}, {σ, 3}, {b, 20000}}, x, MaxIterations → 5000];

  background = gaussian["ParameterTableEntries"][[4, 1]];

  curvemax = Quiet[FindMaximum[gaussian[x], x][[1]]];
  xmax = x /. Quiet[FindMaximum[gaussian[x], x][[2]]];

  sides = Quiet@Solve[gaussian[x] == (curvemax - background) / 2 + background];
  p1 = Re[x /. sides[[1]]];
  p2 = Re[x /. sides[[2]]];
  fwhm = Abs[p2 - p1];
  {Show[Plot[gaussian[x], {x, 1, 8}, PlotStyle → Directive[{Black, AbsoluteThickness[5]}], PlotTheme -> "Monochrome",
     GridLines → Automatic, BaseStyle → {FontSize → 25}, Frame → True],
    ListPlot[{{p1, (curvemax - background) / 2 + background}, {p2, (curvemax - background) / 2 + background}},
     PlotStyle → Directive[{Red, AbsoluteThickness[5]}], Joined → True]], xmax, fwhm, gaussian}
 ]
```

Simple 2 shell model

model function

```
shellmodel[bundle_, approxcenter_, approxr_, dx_, dz_, dr_, vθ1_, vθ2_, vxbulk_, vzbulk_, vax_] :=
 Quiet[Module[{radius, ehat, rvec, r, θhat, axialvelocity, rotationalvelocity, velocity, lineint, sightlinedistances,
    sightlineangles, Lpi, lineintegration, normlineintvalues},
   radius = approxr + dr;
   ehat = unitvec[centerzero, approxcenter + {dx, 0, dz}];
   rvec[{x_, y_, z_}] = ({x, y, z} - centerzero) - (({x, y, z} - centerzero).ehat) * ehat;
   r[{x_, y_, z_}] = norm[rvec[{x, y, z}]];
   θhat[{x_, y_, z_}] = Simplify[Cross[rvec[{x, y, z}], ehat] / norm[Cross[rvec[{x, y, z}], ehat]]];

   axialvelocity[{x_, y_, z_}] =
    Simplify[Piecewise[{{vax * Exp[r[{x, y, z}]^4 / (r[{x, y, z}]^4 - radius^4)] * ehat, r[{x, y, z}] ≤ radius},
       {0 * ehat, r[{x, y, z}] > radius}}]];
   rotationalvelocity[{x_, y_, z_}] =
    Simplify[Piecewise[{{-vθ1 * θhat[{x, y, z}] + {vxbulk, 0, vzbulk}, r[{x, y, z}] ≤ radius / 2},
       {-vθ2 * θhat[{x, y, z}] + {vxbulk, 0, vzbulk}, radius / 2 < r[{x, y, z}] ≤ radius},
       {0 * θhat[{x, y, z}], r[{x, y, z}] > radius}}]];
   velocity[{x_, y_, z_}] = axialvelocity[{x, y, z}] + rotationalvelocity[{x, y, z}];

   (*Show[sights,VectorPlot3D[velocity[{x,y,z}],{x,-.1,-0.5},{y,-.615,.433},{z,-0.1,0.1},VectorPoints→25,
     VectorStyle→Arrowheads[0.01]],Graphics3D[Line[{centerzero,centerend}]]]*)

   sightlinedistances =
    Table[Abs[Dot[Cross[unitvec[bundle[[1]], bundle[[1 + i]]], ehat] / norm[Cross[unitvec[bundle[[1]], bundle[[1 + i]]], ehat]],
       bundle[[1]] - centerzero]], {i, 1, Length[bundle] - 1}];
   (*min distance between sightline and axis line of jet model*)
   sightlineangles = Table[ (Abs[ArcCos[Dot[unitvec[bundle[[1]], bundle[[1 + i]]], ehat]] - π / 2), {i, 1, Length[bundle] - 1}];
   (*angle between sightline and ehat  - π/2*)
   Lpi = Re[2 * Sqrt[radius^2 - sightlinedistances^2] / Cos[sightlineangles]];
   (*length of sightlines through plasma model*)


   lineintegration[{p1x_, p1y_, p1z_}, {p2x_, p2y_, p2z_}] := Quiet[
     NIntegrate[
      Dot[velocity[{p1x, p1y, p1z} + unitvec[{p1x, p1y, p1z}, {p2x, p2y, p2z}] * l], unitvec[{p1x, p1y, p1z}, {p2x, p2y, p2z}]],
      {l, 0, norm[{p1x, p1y, p1z} - {p2x, p2y, p2z}]}, Method → {Automatic, "SymbolicProcessing" → 0}]
    ];


   normlineintvalues = Table[lineintegration[bundle[[1]], bundle[[1 + i]]] / Lpi[[i]] * 10^-3, {i, 1, Length[bundle] - 1}] /.
     {Indeterminate → 0, ComplexInfinity → 0};

   ListPlot[normlineintvalues /. Indeterminate → 0, Joined → True, PlotStyle → Directive[Dashing[0.02], Red, AbsoluteThickness[5]]]
  ]]
```

```
Abelinvert[x_, z_, bundle_, data_] :=
 Module[{sightlines, bundlepoints, ypointvals, centerpoints, centerslope, slopes, slopesabove, slopesbelow, radabove,
   radbelow, sabove, sbelow, Labove, Mabove, LTIabove, MTIabove, Lbelow, Mbelow, LTIbelow, MTIbelow, dataabove,
   databelow, σbvabove, σbvbelow, vabove, eabove, ϵnumabove, ϵdenabove, vaboveerror, vbelow, ebelow, ϵnumbelow,
   ϵdenbelow, vbelowerror, abovetable, abovetable2, belowtable, belowtable2, above, below, toplines, bottomlines},
  sightlines = Length[bundles[[bundle]]] - 1; (*number of sightlines*)
  bundlepoints = bundles[[bundle]];
  (*list of the bundle points that defines line of sight, starts with footpoint, length equalt to sightlines+1*)
  ypointvals = Table[minimizer[bundlepoints[[1]], bundlepoints[[i]], x, z], {i, 2, Length[bundlepoints]}];
  (*assigns the y value to the model center location that minimizes distance to sightline*)
  centerpoints = Table[{x, ypointvals[[i]], z}, {i, 1, Length[ypointvals]}];
  (*creates a list of centerpoints for each line of sight, different axial location but same position in toroidal plane*)
  centerslope = (z - bundlepoints[[1, 3]]) / (x - bundlepoints[[1, 1]]);
  (*calculates the slope from the bundle footpoint to the model centerpoint*)
  slopes = Table[(bundlepoints[[i, 3]] - bundlepoints[[1, 3]]) / (bundlepoints[[i, 1]] - bundlepoints[[1, 1]]),
    {i, 2, Length[bundlepoints]}];
  (*calculates slope of each fiber line of sight to see if it is above or below the model centerpoint*)
  slopesabove = Reverse[Position[slopes, _?(# > centerslope &)]] // Flatten;
  (*index positions of lines of sight above center, in order of increasing radii*)
  slopesbelow = Position[slopes, _?(# < centerslope &)] // Flatten;
  (*index positions of lines of sighte below center*)

  radabove = Table[pointlinedistance[bundlepoints[[1]], bundlepoints[[1 + slopesabove[[i]]]], centerpoints[[slopesabove[[i]]]]],
    {i, 1, Length[slopesabove]}];
  (*list of tangential radii (INCREASING) for lines of sight above center of model*)
  radbelow = Table[pointlinedistance[bundlepoints[[1]], bundlepoints[[1 + slopesbelow[[i]]]], centerpoints[[slopesbelow[[i]]]]],
    {i, 1, Length[slopesbelow]}];
  (*list of tangential radii (INCREASING) for lines of sight below center of model*)

  sabove = Table[(radabove[[i]] + radabove[[i + 1]]) / 2, {i, 1, Length[radabove] - 1}];
  If[Length[slopesabove] > 1, AppendTo[sabove, Last[radabove] + (Last[radabove] - Last[sabove])]];
  If[Length[slopesabove] == 1, AppendTo[sabove, radabove[[1]] * 2]]; (*outer radii of the shells*)
  sbelow = Table[(radbelow[[i]] + radbelow[[i + 1]]) / 2, {i, 1, Length[radbelow] - 1}];
  If[Length[slopesbelow] > 1, AppendTo[sbelow, Last[radbelow] + (Last[radbelow] - Last[sbelow])]];
  If[Length[slopesbelow] == 1, AppendTo[sbelow, radbelow[[1]] * 2]]; (*outer radii of the shells*)

Labove = Table[
  Which[sabove[[j]] > radabove[[i]] && i == j, 2 * Sqrt[sabove[[j]]^2 - radabove[[i]]^2],
   sabove[[j]] > radabove[[i]] && i != j,
   ((2 * Sqrt[sabove[[j]]^2 - radabove[[i]]^2]) - (2 * Sqrt[sabove[[j - 1]]^2 - radabove[[i]]^2])),
   sabove[[j]] < radabove[[i]] , 0], {i, 1, Length[radabove]}, {j, 1, Length[sabove]}];
(*Length matrix*)
Mabove = Table[
  Which[sabove[[j]] > radabove[[i]] && i == j, 2 * Sqrt[sabove[[j]]^2 - radabove[[i]]^2] * radabove[[i]] / radabove[[j]],
   sabove[[j]] > radabove[[i]] && i != j,
   ((2 * Sqrt[sabove[[j]]^2 - radabove[[i]]^2]) - (2 * Sqrt[sabove[[j - 1]]^2 - radabove[[i]]^2])) *
    radabove[[i]] / radabove[[j]],
   sabove[[j]] < radabove[[i]] , 0], {i, 1, Length[radabove]}, {j, 1, Length[sabove]}];
(*length matrix with cosθ = ri/rj term aded*)

LTIabove = Inverse[Transpose[Labove]]; (*inverse transpose of length matrix*)
MTIabove = Inverse[Transpose[Mabove]];


Lbelow = Table[
  Which[sbelow[[j]] > radbelow[[i]] && i == j, 2 * Sqrt[sbelow[[j]]^2 - radbelow[[i]]^2],
   sbelow[[j]] > radbelow[[i]] && i != j,
   ((2 * Sqrt[sbelow[[j]]^2 - radbelow[[i]]^2]) - (2 * Sqrt[sbelow[[j - 1]]^2 - radbelow[[i]]^2])),
   sbelow[[j]] < radbelow[[i]] , 0], {i, 1, Length[radbelow]}, {j, 1, Length[sbelow]}];
Mbelow = Table[
  Which[sbelow[[j]] > radbelow[[i]] && i == j, 2 * Sqrt[sbelow[[j]]^2 - radbelow[[i]]^2 * radbelow[[i]] / radbelow[[j]]],
   sbelow[[j]] > radbelow[[i]] && i != j,
   ((2 * Sqrt[sbelow[[j]]^2 - radbelow[[i]]^2]) - (2 * Sqrt[sbelow[[j - 1]]^2 - radbelow[[i]]^2])) *
    radbelow[[i]] / radbelow[[j]],
   sbelow[[j]] < radbelow[[i]] , 0], {i, 1, Length[radbelow]}, {j, 1, Length[sbelow]}];

LTIbelow = Inverse[Transpose[Lbelow]];
MTIbelow = Inverse[Transpose[Mbelow]];
```

```
dataabove = Table[data[[slopesabove[[i]]]], {i, 1, Length[slopesabove]}];
(*in order of increasing radius*)
databelow = Table[data[[slopesbelow[[i]]]], {i, 1, Length[slopesbelow]}];
(*in order of increasiing radius*)

σbvabove = Abs[dataabove[[ ;; , 1]] * dataabove[[ ;; , 3]]] *
  Sqrt[(dataabove[[ ;; , 2]] / dataabove[[ ;; , 1]])^2 + (dataabove[[ ;; , 4]] / dataabove[[ ;; , 3]])^2];
(*σ for brightness * line integrated velocity*)
σbvbelow = Abs[databelow[[ ;; , 1]] * databelow[[ ;; , 3]]] *
  Sqrt[(databelow[[ ;; , 2]] / databelow[[ ;; , 1]])^2 + (databelow[[ ;; , 4]] / databelow[[ ;; , 3]])^2];
(*σ for brightness * line integrated velocity*)

vabove = Table[(Sum[MTIabove[[i, j]] * dataabove[[i, 1]] * dataabove[[i, 3]], {i, 1, Length[slopesabove]}]) /
    (Sum[LTIabove[[i, j]] * dataabove[[i, 3]], {i, 1, Length[slopesabove]}]), {j, 1, Length[slopesabove]}];
eabove = Table[(Sum[LTIabove[[i, j]] * dataabove[[i, 3]], {i, 1, Length[slopesabove]}]), {j, 1, Length[slopesabove]}];
enumabove = Table[(Sqrt[Sum[(MTIabove[[i, j]] * σbvabove[[i]])^2, {i, 1, Length[slopesabove]}]]) /
    (Sum[MTIabove[[i, j]] * dataabove[[i, 1]] * dataabove[[i, 3]], {i, 1, Length[slopesabove]}]), {j, 1, Length[slopesabove]}];
edenabove = Table[(Sqrt[Sum[(LTIabove[[i, j]] * dataabove[[i, 4]])^2, {i, 1, Length[slopesabove]}]]) /
    (Sum[LTIabove[[i, j]] * dataabove[[i, 3]], {i, 1, Length[slopesabove]}]), {j, 1, Length[slopesabove]}];
vaboveerror = Abs[vabove] * Sqrt[enumabove^2 + edenabove^2];

vbelow = Table[(Sum[MTIbelow[[i, j]] * databelow[[i, 1]] * databelow[[i, 3]], {i, 1, Length[slopesbelow]}]) /
    (Sum[LTIbelow[[i, j]] * databelow[[i, 3]], {i, 1, Length[slopesbelow]}]), {j, 1, Length[slopesbelow]}];
ebelow = Table[(Sum[LTIbelow[[i, j]] * databelow[[i, 3]], {i, 1, Length[slopesbelow]}]), {j, 1, Length[slopesbelow]}];
enumbelow = Table[(Sqrt[Sum[(MTIbelow[[i, j]] * σbvbelow[[i]])^2, {i, 1, Length[slopesbelow]}]]) /
    (Sum[MTIbelow[[i, j]] * databelow[[i, 1]] * databelow[[i, 3]], {i, 1, Length[slopesbelow]}]), {j, 1, Length[slopesbelow]}];
edenbelow = Table[(Sqrt[Sum[(LTIbelow[[i, j]] * databelow[[i, 4]])^2, {i, 1, Length[slopesbelow]}]]) /
    (Sum[LTIbelow[[i, j]] * databelow[[i, 3]], {i, 1, Length[slopesbelow]}]), {j, 1, Length[slopesbelow]}];
vbelowerror = Abs[vbelow] * Sqrt[enumbelow^2 + edenbelow^2];

abovetable = Table[{{-1 * radabove[[i]], vabove[[i]]}, ErrorBar[vaboveerror[[i]]]}, {i, 1, Length[slopesabove]}];
abovetable2 = Table[{{-1 * radabove[[i]], eabove[[i]]}, ErrorBar[edenabove[[i]] * eabove[[i]]]}, {i, 1, Length[slopesabove]}];

belowtable = Table[{{radbelow[[i]], vbelow[[i]]}, ErrorBar[vbelowerror[[i]]]}, {i, 1, Length[slopesbelow]}];
belowtable2 = Table[{{radbelow[[i]], ebelow[[i]]}, ErrorBar[edenbelow[[i]] * ebelow[[i]]]}, {i, 1, Length[slopesbelow]}];

above = Show[Table[Graphics[{Dashed, Circle[{0, 0}, radabove[[i]], {0, π}]}], {i, 1, Length[radabove]}],
  Table[Graphics[{Red, Circle[{0, 0}, sabove[[i]], {0, π}]}], {i, 1, Length[sabove]}]];

below = Show[Table[Graphics[{Dashed, Circle[{0, 0}, radbelow[[i]], {π, 2π}]}], {i, 1, Length[radbelow]}],
  Table[Graphics[{Red, Circle[{0, 0}, sbelow[[i]], {π, 2π}]}], {i, 1, Length[sbelow]}]];

toplines = Table[Graphics[Line[{{-0.5, 0}, {0, radabove[[i]]}}]], {i, 1, Length[slopesabove]}];
bottomlines = Table[Graphics[Line[{{-0.5, 0}, {0, -radbelow[[i]]}}]], {i, 1, Length[slopesbelow]}];

 {Show[above, below, toplines, bottomlines, ImageSize → 600],
  ErrorListPlot[{abovetable, belowtable} /. {} → {∅}, Joined → True, Frame → True,
   PlotStyle → {Directive[Red, Thickness[0.01]], Directive[Red, Thickness[0.01]]}, PlotMarkers → {Graphics@{Red, Disk[]}, 0.05},
   PlotRange → {{-0.12, 0.12}, {-100, 120}}, BaseStyle → FontSize → 35, FrameStyle → Thick,
   GridLines → {{-.1, -.05, 0, 0.05, 0.1}, {-100, -80, -60, -40, -20, 0, 20, 40, 60, 80, 100}},
   FrameTicks -> {{{-100, -80, -60, -40, -20, 0, 20, 40, 60, 80, 100}, None}, {{-.1, -.05, 0, 0.05, 0.1}, None}},
   GridLinesStyle → Dashed, ImageSize → 800], ErrorListPlot[{abovetable2, belowtable2} /. {} → {∅}, Joined → True,
   ImageSize → 800, PlotStyle → {Directive[Red, Thickness[0.01]], Directive[Red, Thickness[0.01]]},
   PlotMarkers → {Graphics@{Red, Disk[]}, 0.05}], abovetable, belowtable}


](*normal abel inversion using the total area under the binned intensity curves*)
```

APPENDIX F
MAGNETIC FIELD CODES

The following python code was to analyze results from MPA 1 during the first campaign, prior to the calibration and implementation of MPA2 and 3.

```
import MDSplus as mds
import numpy as np
from numpy.linalg import inv
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.optimize import minimize
import scipy.fftpack as fft
from scipy.fftpack import rfft,irfft,fftfreq
import scipy.integrate as integrate
from math import factorial
from scipy.signal import butter, lfilter
from scipy.signal import freqz
from scipy import special
from scipy.interpolate import griddata
from scipy import stats
%matplotlib
Using matplotlib backend: Qt4Agg

def multerror(val1,val2,sigma1,sigma2):
    sigma = abs(val1*val2)*np.sqrt((sigma1/val1)**2+(sigma2/val2)**2)
    return sigma


def addsuberror(sigma1,sigma2):
    sigma = np.sqrt(sigma1**2+sigma2**2)
    return sigma


def det(a,b,c,d,sa,sb,sc,sd):
    ans = a*d-b*c
    e1 = multerror(a,d,sa,sd)
    e2 = multerror(b,c,sb,sc)
    error = addsuberror(e1,e2)
    return ans,error


def matrixinverse(matrix,sigmamatrix):
    realdet = np.linalg.det(matrix)
```

```
    a = matrix[0][0]
    b = matrix[0][1]
    c = matrix[0][2]
    d = matrix[1][0]
    e = matrix[1][1]
    f = matrix[1][2]
    g = matrix[2][0]
    h = matrix[2][1]
    k = matrix[2][2]

    aa = sigmamatrix[0][0]
    bb = sigmamatrix[0][1]
    cc = sigmamatrix[0][2]
    dd = sigmamatrix[1][0]
    ee = sigmamatrix[1][1]
    ff = sigmamatrix[1][2]
    gg = sigmamatrix[2][0]
    hh = sigmamatrix[2][1]
    kk = sigmamatrix[2][2]


    determinant = a*det(e,f,h,k,ee,ff,hh,kk)[0] - b*det(d,f,g,k,dd,ff,gg,kk)[
0] + c*det(d,e,g,h,dd,ee,gg,hh)[0]

    e1 = multerror(a,det(e,f,h,k,ee,ff,hh,kk)[0],aa,det(e,f,h,k,ee,ff,hh,kk)[
1])
    e2 = multerror(b,det(d,f,g,k,dd,ff,gg,kk)[0],bb,det(d,f,g,k,dd,ff,gg,kk)[
1])
    e3 = multerror(c,det(d,e,g,h,dd,ee,gg,hh)[0],cc,det(d,e,g,h,dd,ee,gg,hh)[
1])

    deterror = np.sqrt(e1**2+e2**2+e3**2)

    detpererror = deterror/determinant

    minv = np.array([[1/determinant*det(e,f,h,k,ee,ff,hh,kk)[0],1/determinant
*det(c,b,k,h,cc,bb,kk,hh)[0],1/determinant*det(b,c,e,f,bb,cc,ee,ff)[0]],
                            [1/determinant*det(f,d,k,g,ff,dd,kk,gg)[0],1/determ
inant*det(a,c,g,k,aa,cc,gg,kk)[0],1/determinant*det(c,a,f,d,cc,aa,ff,dd)[0]],
                            [1/determinant*det(d,e,g,h,dd,ee,gg,hh)[0],1/determ
inant*det(b,a,h,g,bb,aa,hh,gg)[0],1/determinant*det(a,b,d,e,aa,bb,dd,ee)[0]]]
)
```

```
    sigmainv = np.array([[multerror(1/determinant,det(e,f,h,k,ee,ff,hh,kk)[0]
,1/determinant*detpererror,det(e,f,h,k,ee,ff,hh,kk)[1]),
                  multerror(1/determinant,det(c,b,k,h,cc,bb,kk,hh)[0],1/determ
inant*detpererror,det(c,b,k,h,cc,bb,kk,hh)[1]),
                  multerror(1/determinant,det(c,b,k,h,cc,bb,kk,hh)[0],1/determ
inant*detpererror,det(c,b,k,h,cc,bb,kk,hh)[1])],
                 [multerror(1/determinant,det(f,d,k,g,ff,dd,kk,gg)[0],1/determ
inant*detpererror,det(f,d,k,g,ff,dd,kk,gg)[1]),
                  multerror(1/determinant,det(a,c,g,k,aa,cc,gg,kk)[0],1/determ
inant*detpererror,det(a,c,g,k,aa,cc,gg,kk)[1]),
                  multerror(1/determinant,det(c,a,f,d,cc,aa,ff,dd)[0],1/determ
inant*detpererror,det(c,a,f,d,cc,aa,ff,dd)[1])],
                 [multerror(1/determinant,det(d,e,g,h,dd,ee,gg,hh)[0],1/determ
inant*detpererror,det(d,e,g,h,dd,ee,gg,hh)[1]),
                  multerror(1/determinant,det(b,a,h,g,bb,aa,hh,gg)[0],1/determ
inant*detpererror,det(b,a,h,g,bb,aa,hh,gg)[1]),
                  multerror(1/determinant,det(a,b,d,e,aa,bb,dd,ee)[0],1/determ
inant*detpererror,det(a,b,d,e,aa,bb,dd,ee)[1])]])

    return minv, sigmainv

def butter_bandpass(lowcut,highcut,fs,order=5):
    nyq=0.5*fs
    low=lowcut/nyq
    high=highcut/nyq
    b, a = butter(order,[low,high],btype='band')
    return b, a

def butter_bandpass_filter(data,lowcut,highcut,fs,order=5):
    b, a = butter_bandpass(lowcut,highcut,fs,order=order)
    y=lfilter(b, a, data)
    return y

def savitzky_golay(y, window_size, order, deriv=0, rate=1):
    try:
        window_size = np.abs(np.int(window_size))
        order = np.abs(np.int(order))
    except ValueError, msg:
        raise ValueError("window_size and order have to be of type int")
    if window_size % 2 != 1 or window_size < 1:
        raise TypeError("window_size size must be a positive odd number")
    if window_size < order + 2:
        raise TypeError("window_size is too small for the polynomials order")
```

```
    order_range = range(order+1)
    half_window = (window_size -1) // 2
    # precompute coefficients
    b = np.mat([[k**i for i in order_range] for k in range(-half_window, half
_window+1)])
    m = np.linalg.pinv(b).A[deriv] * rate**deriv * factorial(deriv)
    # pad the signal at the extremes with
    # values taken from the signal itself
    firstvals = y[0] - np.abs( y[1:half_window+1][::-1] - y[0] )
    lastvals = y[-1] + np.abs(y[-half_window-1:-1][::-1] - y[-1])
    y = np.concatenate((firstvals, y, lastvals))
    return np.convolve( m[::-1], y, mode='valid')


def calresponseplot(shot):

    tree=mds.Tree('proto_tree',shot)

    tnode=tree.getNode('\VTIMEVALUES_RCARD1CH0')
    inode=tree.getNode('\VDATAVALUES_RCARD1CH0')
    vxnode=tree.getNode('\VDATAVALUES_RCARD1CH2')
    vynode=tree.getNode('\VDATAVALUES_RCARD1CH6')
    vznode=tree.getNode('\VDATAVALUES_RCARD1CH4')

    mds_time=tnode.getData()
    mds_i=inode.getData()
    mds_vx=vxnode.getData()
    mds_vy=vynode.getData()
    mds_vz=vznode.getData()

    t0=np.array(mds_time)
    rawi=(np.array(mds_i))/0.1
    rawvx=np.array(mds_vx)
    rawvy=np.array(mds_vy)
    rawvz=np.array(mds_vz)


    #sometimes at the end of the shot the voltage jumps, so I cut off the las
t 1/10th of the data
    length=len(t0)
    stop=int(0.9*length)
    time=t0[:stop]
    current=rawi[:stop]
    vx=rawvx[:stop]
```

```
    vy=rawvy[:stop]
    vz=rawvz[:stop]


    n=len(time)
    ts=time[1]-time[0]
    current_fft = fft.fft(current)
    freqs=fft.fftfreq(int(n),ts)
    abscurrent_fft=abs(current_fft[1:n/2])
    maxvalindex=np.where(abscurrent_fft[:n]==np.amax(abscurrent_fft))[0][0]+1

    def my_sin(x, freq, amplitude, phase, offset):
        return np.sin(x * freq + phase) * amplitude + offset

#    ci=0.95 #confidence interval
#    pp = (1.+ci)/2.  # convert to percentile point of normal distribution.
    nstd = 20 #stats.norm.ppf(pp) #number of standard deviations


    guess_freq=2*np.pi*freqs[maxvalindex]
    guess_phase=0
    guess_ioffset=0.5*(np.amax(current)+np.amin(current))
    guess_iamp=0.5*(np.amax(current)-np.amin(current))
    current_guess=[guess_freq, guess_iamp, guess_phase, guess_ioffset]
    ifit=curve_fit(my_sin,time,current,p0=current_guess)
    current_fit=my_sin(time,*ifit[0])
    shifted_current=current-ifit[0][3]
    shifted_current_fit=current_fit-ifit[0][3]
    sigmaAcurrent = np.sqrt(np.diag(ifit[1]))[1]*nstd


    guess_vxoffset=0.5*(np.amax(vx)+np.amin(vx))
    guess_vxamp=0.5*(np.amax(vx)-np.amin(vx))
    vx_guess=[guess_freq, guess_vxamp, guess_phase, guess_vxoffset]
    vxfit=curve_fit(my_sin,time,vx,p0=vx_guess)
    vx_fit=my_sin(time,*vxfit[0])
    shifted_vx=vx-vxfit[0][3]
    shifted_vx_fit=vx_fit-vxfit[0][3]
    sigmaAvx = np.sqrt(np.diag(vxfit[1]))[1]*nstd
    sigmaFvx = np.sqrt(np.diag(vxfit[1]))[0]*nstd


    guess_vyoffset=0.5*(np.amax(vy)+np.amin(vy))
    guess_vyamp=0.5*(np.amax(vy)-np.amin(vy))
```

```
    vy_guess=[guess_freq, guess_vyamp, guess_phase, guess_vyoffset]
    vyfit=curve_fit(my_sin,time,vy,p0=vy_guess)
    vy_fit=my_sin(time,*vyfit[0])
    shifted_vy=vy-vyfit[0][3]
    shifted_vy_fit=vy_fit-vyfit[0][3]
    sigmaAvy = np.sqrt(np.diag(vyfit[1]))[1]*nstd
    sigmaFvy = np.sqrt(np.diag(vyfit[1]))[0]*nstd


    guess_vzoffset=0.5*(np.amax(vz)+np.amin(vz))
    guess_vzamp=0.5*(np.amax(vz)-np.amin(vz))
    vz_guess=[guess_freq, guess_vzamp, guess_phase, guess_vzoffset]
    vzfit=curve_fit(my_sin,time,vz,p0=vz_guess)
    vz_fit=my_sin(time,*vzfit[0])
    shifted_vz=vz-vzfit[0][3]
    shifted_vz_fit=vz_fit-vzfit[0][3]
    sigmaAvz = np.sqrt(np.diag(vzfit[1]))[1]*nstd
    sigmaFvz = np.sqrt(np.diag(vzfit[1]))[0]*nstd


    plt.figure(facecolor='white',figsize=(30,20))
    plt.subplot(1,4,1)
    plt.plot(time*1e6,shifted_current_fit,color='r',linewidth=3)
#    plt.fill_between(time*1e6,shifted_current_fit-sigmaAcurrent,shifted_curr
ent_fit+sigmaAcurrent,facecolor='orange')
    plt.plot(time*1e6,shifted_current,color='black')
    plt.tick_params(labelsize=30)
    plt.ticklabel_format()
    plt.axis([0,10,-0.2,0.2])


    plt.subplot(1,4,2)
    plt.plot(time*1e6,shifted_vx_fit,color='r',linewidth=3)
#    plt.fill_between(time*1e6,shifted_vx_fit-sigmaAvx,shifted_vx_fit+sigmaAv
x,facecolor='orange')
    plt.plot(time*1e6,shifted_vx,color='b')
    plt.tick_params(labelsize=30)
    plt.axis([0,10,-0.002,0.002])


    plt.subplot(1,4,3)
    plt.plot(time*1e6,shifted_vy,color='g')
#    plt.fill_between(time*1e6,shifted_vy_fit-sigmaAvx,shifted_vy_fit+sigmaAv
x,facecolor='orange')
```

```
    plt.plot(time*1e6,shifted_vy_fit,color = 'r',linewidth=3)
    plt.tick_params(labelsize=30)
    plt.axis([0,10,-0.002,0.002])

    plt.subplot(1,4,4)
    plt.plot(time*1e6,shifted_vz,color='r')
#    plt.fill_between(time*1e6,shifted_vz_fit-sigmaAvx,shifted_vz_fit+sigmaAv
x,facecolor='orange')
    plt.plot(time*1e6,shifted_vz_fit,color='r',linewidth=3)
    plt.tick_params(labelsize=30)
    plt.axis([0,10,-0.002,0.002])

    plt.tight_layout()
    plt.show()

    print nstd


 def calibrate(shot,direction):
    tree=mds.Tree('proto_tree',shot)

    tnode=tree.getNode('\VTIMEVALUES_RCARD1CH0')
    inode=tree.getNode('\VDATAVALUES_RCARD1CH0')
    vxnode=tree.getNode('\VDATAVALUES_RCARD1CH2')
    vynode=tree.getNode('\VDATAVALUES_RCARD1CH6')
    vznode=tree.getNode('\VDATAVALUES_RCARD1CH4')

    mds_time=tnode.getData()
    mds_i=inode.getData()
    mds_vx=vxnode.getData()
    mds_vy=vynode.getData()
    mds_vz=vznode.getData()

    t0=np.array(mds_time)
    rawi=(np.array(mds_i))/0.1
    rawvx=np.array(mds_vx)
    rawvy=np.array(mds_vy)
    rawvz=np.array(mds_vz)


    #sometimes at the end of the shot the voltage jumps, so I cut off the las
t 1/10th of the data
    length=len(t0)
    stop=int(0.9*length)
```

```
time=t0[:stop]
current=rawi[:stop]
vx=rawvx[:stop]
vy=rawvy[:stop]
vz=rawvz[:stop]


n=len(time)
ts=time[1]-time[0]
current_fft = fft.fft(current)
freqs=fft.fftfreq(int(n),ts)
abscurrent_fft=abs(current_fft[1:n/2])
maxvalindex=np.where(abscurrent_fft[:n]==np.amax(abscurrent_fft))[0][0]+1

def my_sin(x, freq, amplitude, phase, offset):
    return np.sin(x * freq + phase) * amplitude + offset

nstd = 20

guess_freq=2*np.pi*freqs[maxvalindex]
guess_phase=0
guess_ioffset=0.5*(np.amax(current)+np.amin(current))
guess_iamp=0.5*(np.amax(current)-np.amin(current))
current_guess=[guess_freq, guess_iamp, guess_phase, guess_ioffset]
ifit=curve_fit(my_sin,time,current,p0=current_guess)
current_fit=my_sin(time,*ifit[0])
shifted_current=current-ifit[0][3]
shifted_current_fit=current_fit-ifit[0][3]
sigmaAcurrent = np.sqrt(np.diag(ifit[1]))[1]


guess_vxoffset=0.5*(np.amax(vx)+np.amin(vx))
guess_vxamp=0.5*(np.amax(vx)-np.amin(vx))
vx_guess=[guess_freq, guess_vxamp, guess_phase, guess_vxoffset]
vxfit=curve_fit(my_sin,time,vx,p0=vx_guess)
vx_fit=my_sin(time,*vxfit[0])
shifted_vx=vx-vxfit[0][3]
shifted_vx_fit=vx_fit-vxfit[0][3]
sigmaAvx = np.sqrt(np.diag(vxfit[1]))[1]
sigmaFvx = np.sqrt(np.diag(vxfit[1]))[0]


guess_vyoffset=0.5*(np.amax(vy)+np.amin(vy))
guess_vyamp=0.5*(np.amax(vy)-np.amin(vy))
```

```
vy_guess=[guess_freq, guess_vyamp, guess_phase, guess_vyoffset]
vyfit=curve_fit(my_sin,time,vy,p0=vy_guess)
vy_fit=my_sin(time,*vyfit[0])
shifted_vy=vy-vyfit[0][3]
shifted_vy_fit=vy_fit-vyfit[0][3]
sigmaAvy = np.sqrt(np.diag(vyfit[1]))[1]
sigmaFvy = np.sqrt(np.diag(vyfit[1]))[0]


guess_vzoffset=0.5*(np.amax(vz)+np.amin(vz))
guess_vzamp=0.5*(np.amax(vz)-np.amin(vz))
vz_guess=[guess_freq, guess_vzamp, guess_phase, guess_vzoffset]
vzfit=curve_fit(my_sin,time,vz,p0=vz_guess)
vz_fit=my_sin(time,*vzfit[0])
shifted_vz=vz-vzfit[0][3]
shifted_vz_fit=vz_fit-vzfit[0][3]
sigmaAvz = np.sqrt(np.diag(vzfit[1]))[1]
sigmaFvz = np.sqrt(np.diag(vzfit[1]))[0]


def my_cos(x, freq, amplitude, phase):
    return np.sin(x * freq + phase) * amplitude

vx_int=my_cos(time,vxfit[0][0],-(vxfit[0][1]/vxfit[0][0]),vxfit[0][2])
vy_int=my_cos(time,vyfit[0][0],-(vyfit[0][1]/vyfit[0][0]),vyfit[0][2])
vz_int=my_cos(time,vzfit[0][0],-(vzfit[0][1]/vzfit[0][0]),vzfit[0][2])



mu0=np.pi*4e-7
nx=4
ny=4
nz=5
ax=.0187
sigmaax=0.0002

ay=.02287
sigmaay=0.00012

az=.02750
sigmaaz=0.00015

if direction == 'x':
    bapplied=(4./5.)**(3./2.)*mu0*nx*shifted_current_fit/ax
```

```
        bsigma = (4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaax/ax)**
2+(sigmaAcurrent/ifit[0][1])**2)
    elif direction == 'y':
        bapplied=(4./5.)**(3./2.)*mu0*ny*shifted_current_fit/ay
        bsigma =  (4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaay/ay)*
*2+(sigmaAcurrent/ifit[0][1])**2)
    elif direction == 'z':
        bapplied=(4./5.)**(3./2.)*mu0*nz*shifted_current_fit/az
        bsigma = (4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaaz/az)**
2+(sigmaAcurrent/ifit[0][1])**2)

    bapplied_amplitude = (np.amax(bapplied)-np.amin(bapplied))/2


    grad_vx_int=np.gradient(vx_int)
    grad_vy_int=np.gradient(vy_int)
    grad_vz_int=np.gradient(vz_int)
    grad_current=np.gradient(current_fit)



    if grad_current[0]>0:
        index1=np.where(grad_current<0)[0][0]
    elif grad_current[0]<0:
        index1=np.where(grad_current>0)[0][0]


    if grad_vx_int[index1]>0:
        index2=index1+np.where(grad_vx_int[index1:]<0)[0][0]
    elif grad_vx_int[index1]<0:
        index2=index1+np.where(grad_vx_int[index1:]>0)[0][0]

    if grad_vy_int[index1]>0:
        index3=index1+np.where(grad_vy_int[index1:]<0)[0][0]
    elif grad_vy_int[index1]<0:
        index3=index1+np.where(grad_vy_int[index1:]>0)[0][0]

    if grad_vz_int[index1]>0:
        index4=index1+np.where(grad_vz_int[index1:]<0)[0][0]
    elif grad_vz_int[index1]<0:
        index4=index1+np.where(grad_vz_int[index1:]>0)[0][0]


    xsign = np.sign(vx_int[index2]/bapplied[index1])
```

```
    ysign = np.sign(vy_int[index3]/bapplied[index1])
    zsign = np.sign(vz_int[index4]/bapplied[index1])


    xcalfactor = abs(vxfit[0][1]/vxfit[0][0])/bapplied_amplitude
    ycalfactor = abs(vyfit[0][1]/vyfit[0][0])/bapplied_amplitude
    zcalfactor = abs(vzfit[0][1]/vzfit[0][0])/bapplied_amplitude

    xcalsigma = xcalfactor*np.sqrt((nstd*sigmaAvx/vxfit[0][1])**2+(5*sigmaFvx
/vxfit[0][0])**2+(bsigma/bapplied_amplitude)**2)
    ycalsigma = ycalfactor*np.sqrt((nstd*sigmaAvy/vyfit[0][1])**2+(5*sigmaFvy
/vyfit[0][0])**2+(bsigma/bapplied_amplitude)**2)
    zcalsigma = zcalfactor*np.sqrt((nstd*sigmaAvz/vzfit[0][1])**2+(5*sigmaFvz
/vzfit[0][0])**2+(bsigma/bapplied_amplitude)**2)

    return(xcalfactor,ycalfactor,zcalfactor,xsign,ysign,zsign,xcalsigma,ycals
igma,zcalsigma)



def signmatrix(xshot,yshot,zshot):
    xresponse=calibrate(xshot+1,'x')
    yresponse=calibrate(yshot+1,'y')
    zresponse=calibrate(zshot+1,'z')

    m=[[xresponse[3],yresponse[3],zresponse[3]],[xresponse[4],yresponse[4],zr
esponse[4]],[xresponse[5],yresponse[5],zresponse[5]]]

    return m

def fullmatrix(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')


    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],zresponse[0]*sig
ns[2][0]],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],zresponse[1]*si
gns[2][1]],
          [xresponse[2]*signs[2][0],yresponse[2]*signs[2][1],zresponse[2]*si
gns[2][2]]]
```

```
    m=inv(minv)

    return m

def fulltest(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')


    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],zresponse[0]*sig
ns[2][0]],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],zresponse[1]*si
gns[2][1]],
          [xresponse[2]*signs[2][0],yresponse[2]*signs[2][1],zresponse[2]*si
gns[2][2]]]

    sigmainv = [[xresponse[6],yresponse[6],zresponse[6]],
            [xresponse[7],yresponse[7],zresponse[7]],
            [xresponse[8],yresponse[8],zresponse[8]]]

    m,sigmam = matrixinverse(minv,sigmainv)

    return m, sigmam


#def fullmatrixplus(xshot,yshot,zshot,nfreq):
#    signs=signmatrix(xshot,yshot,zshot)
#
#    xresponse=calibrate(xshot+nfreq,'x')
#    yresponse=calibrate(yshot+nfreq,'y')
#    zresponse=calibrate(zshot+nfreq,'z')
#
#
#    minv=[[(xresponse[0]+xresponse[6])*signs[0][0],(yresponse[0]+yresponse[6
])*signs[0][1],(zresponse[0]+zresponse[6])*signs[2][0]],
#          [(xresponse[1]+xresponse[7])*signs[1][0],(yresponse[1]+yresponse[
7])*signs[1][1],(zresponse[1]+zresponse[7])*signs[2][1]],
#          [(xresponse[2]+xresponse[8])*signs[2][0],(yresponse[2]+yresponse[
8])*signs[2][1],(zresponse[2]+zresponse[8])*signs[2][2]]]
#    m=inv(minv)
#
```

```
#    return m
#
#def fullmatrixminus(xshot,yshot,zshot,nfreq):
#    signs=signmatrix(xshot,yshot,zshot)
#
#    xresponse=calibrate(xshot+nfreq,'x')
#    yresponse=calibrate(yshot+nfreq,'y')
#    zresponse=calibrate(zshot+nfreq,'z')
#
#
#    minv=[[(xresponse[0]-xresponse[6])*signs[0][0],(yresponse[0]-yresponse[6
])*signs[0][1],(zresponse[0]-zresponse[6])*signs[2][0]],
#          [(xresponse[1]-xresponse[7])*signs[1][0],(yresponse[1]-yresponse[
7])*signs[1][1],(zresponse[1]-zresponse[7])*signs[2][1]],
#          [(xresponse[2]-xresponse[8])*signs[2][0],(yresponse[2]-yresponse[
8])*signs[2][1],(zresponse[2]-zresponse[8])*signs[2][2]]]
#    m=inv(minv)
#
#    return m

def reducedmatrix(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')


    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],0],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],0],
          [0,0,zresponse[2]*signs[2][2]]]
    m=inv(minv)

    return m

def reducedtest(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')

    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],1e-20],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],1e-20],
```

```
            [1e-20,1e-20,zresponse[2]*signs[2][2]]]

    sigmainv = [[xresponse[6],yresponse[6],zresponse[6]],
            [xresponse[7],yresponse[7],zresponse[7]],
            [xresponse[8],yresponse[8],zresponse[8]]]

    m,sigmam = matrixinverse(minv,sigmainv)

    return m, sigmam

#def reducedmatrixplus(xshot,yshot,zshot,nfreq):
#    signs=signmatrix(xshot,yshot,zshot)
#
#    xresponse=calibrate(xshot+nfreq,'x')
#    yresponse=calibrate(yshot+nfreq,'y')
#    zresponse=calibrate(zshot+nfreq,'z')
#
#
#    minv=[[(xresponse[0]+xresponse[6])*signs[0][0],(yresponse[0]+yresponse[6
])*signs[0][1],(0+zresponse[6])*signs[2][0]],
#           [(xresponse[1]+xresponse[7])*signs[1][0],(yresponse[1]+yresponse[
7])*signs[1][1],(0+zresponse[7])*signs[2][1]],
#           [(0+xresponse[8])*signs[2][0],(0+yresponse[8])*signs[2][1],(zresp
onse[2]+zresponse[8])*signs[2][2]]]
#    m=inv(minv)
#
#    return m
#
#def reducedmatrixminus(xshot,yshot,zshot,nfreq):
#    signs=signmatrix(xshot,yshot,zshot)
#
#    xresponse=calibrate(xshot+nfreq,'x')
#    yresponse=calibrate(yshot+nfreq,'y')
#    zresponse=calibrate(zshot+nfreq,'z')
#
#
#    minv=[[(xresponse[0]-xresponse[6])*signs[0][0],(yresponse[0]-yresponse[6
])*signs[0][1],(0-zresponse[6])*signs[2][0]],
#           [(xresponse[1]-xresponse[7])*signs[1][0],(yresponse[1]-yresponse[
7])*signs[1][1],(0-zresponse[7])*signs[2][1]],
#           [(0-xresponse[8])*signs[2][0],(0-yresponse[8])*signs[2][1],(zresp
onse[2]-zresponse[8])*signs[2][2]]]
#    m=inv(minv)
#
```

```
#    return m

#def sigmamatrix(xshot,yshot,zshot,nfreq):
#    signs=signmatrix(xshot,yshot,zshot)
#
#    xresponse=calibrate(xshot+nfreq,'x')
#    yresponse=calibrate(yshot+nfreq,'y')
#    zresponse=calibrate(zshot+nfreq,'z')
#
#    minv = [[xresponse[6],yresponse[6],zresponse[6]],
#            [xresponse[7],yresponse[7],zresponse[7]],
#            [xresponse[8],yresponse[8],zresponse[8]]]
#    m=inv(minv)
#
#    return m



def Breconstruct1(shot,direction,xshot,yshot,zshot,nfreq):

    m = fulltest(xshot,yshot,zshot,nfreq)[0]
    msigma= fulltest(xshot,yshot,zshot,nfreq)[1]


    tree=mds.Tree('proto_tree',shot)
    tnode=tree.getNode('\VTIMEVALUES_RCARD1CH0')
    inode=tree.getNode('\VDATAVALUES_RCARD1CH0')
    vxnode=tree.getNode('\VDATAVALUES_RCARD1CH2')
    vynode=tree.getNode('\VDATAVALUES_RCARD1CH6')
    vznode=tree.getNode('\VDATAVALUES_RCARD1CH4')

    mds_time=tnode.getData()
    mds_i=inode.getData()
    mds_vx=vxnode.getData()
    mds_vy=vynode.getData()
    mds_vz=vznode.getData()

    time=np.array(mds_time)
    current=(np.array(mds_i))/0.1
    vx=np.array(mds_vx)
    vy=np.array(mds_vy)
    vz=np.array(mds_vz)
```

```
    n=len(time)
    ts=time[1]-time[0]
    current_fft = fft.fft(current)
    freqs=fft.fftfreq(int(n),ts)
    abscurrent_fft=abs(current_fft[1:n/2])
    maxvalindex=np.where(abscurrent_fft[:n]==np.amax(abscurrent_fft))[0][0]+1

    def my_sin(x, freq, amplitude, phase, offset):
        return np.sin(x * freq + phase) * amplitude + offset


    guess_freq=2*np.pi*freqs[maxvalindex]
    guess_phase=0
    guess_ioffset=0.5*(np.amax(current)+np.amin(current))
    guess_iamp=0.5*(np.amax(current)-np.amin(current))
    current_guess=[guess_freq, guess_iamp, guess_phase, guess_ioffset]
    ifit=curve_fit(my_sin,time,current,p0=current_guess)
    shifted_current = current-ifit[0][3]
    filtered_current = butter_bandpass_filter(shifted_current,200000,23e6,50e
6,order=3)


    guess_vxoffset=0.5*(np.amax(vx)+np.amin(vx))
    guess_vxamp=0.5*(np.amax(vx)-np.amin(vx))
    vx_guess=[guess_freq, guess_vxamp, guess_phase, guess_vxoffset]
    vxfit=curve_fit(my_sin,time,vx,p0=vx_guess)
    shifted_vx=vx-vxfit[0][3]
    filtered_vx=butter_bandpass_filter(shifted_vx,200000,23e6,50e6,order=3)


    guess_vyoffset=0.5*(np.amax(vy)+np.amin(vy))
    guess_vyamp=0.5*(np.amax(vy)-np.amin(vy))
    vy_guess=[guess_freq, guess_vyamp, guess_phase, guess_vyoffset]
    vyfit=curve_fit(my_sin,time,vy,p0=vy_guess)
    shifted_vy=vy-vyfit[0][3]
    filtered_vy=butter_bandpass_filter(shifted_vy,200000,23e6,50e6,order=3)


    guess_vzoffset=0.5*(np.amax(vz)+np.amin(vz))
    guess_vzamp=0.5*(np.amax(vz)-np.amin(vz))
    vz_guess=[guess_freq, guess_vzamp, guess_phase, guess_vzoffset]
    vzfit=curve_fit(my_sin,time,vz,p0=vz_guess)
    shifted_vz=vz-vzfit[0][3]
    filtered_vz=butter_bandpass_filter(shifted_vz,200000,23e6,50e6,order=3)
```

```
    mu0=np.pi*4e-7
    nx=4
    ny=4
    nz=5
    ax=.0187
    ay=.02287
    az=.02750

    if direction == 'x':
        bapplied=(4./5.)**(3./2.)*mu0*nx*filtered_current/ax
    elif direction == 'y':
        bapplied=(4./5.)**(3./2.)*mu0*ny*filtered_current/ay
    elif direction == 'z':
        bapplied=(4./5.)**(3./2.)*mu0*nz*filtered_current/az


    tstartindex=0
    tspan=time[tstartindex:]

    realint_vx=integrate.cumtrapz(filtered_vx[tstartindex:],tspan,initial=0)
    realint_vy=integrate.cumtrapz(filtered_vy[tstartindex:],tspan,initial=0)
    realint_vz=integrate.cumtrapz(filtered_vz[tstartindex:],tspan,initial=0)


    bxappreal = m[0][0]*realint_vx+m[0][1]*realint_vy+m[0][2]*realint_vz
    byappreal = m[1][0]*realint_vx+m[1][1]*realint_vy+m[1][2]*realint_vz
    bzappreal = m[2][0]*realint_vx+m[2][1]*realint_vy+m[2][2]*realint_vz

    bxsigma = np.sqrt((msigma[0][0]*realint_vx)**2+(msigma[0][1]*realint_vy)*
*2+(msigma[0][2]*realint_vz)**2)
    bysigma = np.sqrt((msigma[1][0]*realint_vx)**2+(msigma[1][1]*realint_vy)*
*2+(msigma[1][2]*realint_vz)**2)
    bzsigma = np.sqrt((msigma[2][0]*realint_vx)**2+(msigma[2][1]*realint_vy)*
*2+(msigma[2][2]*realint_vz)**2)

    bxplus = bxappreal+bxsigma
    bxminus = bxappreal-bxsigma

    byplus = byappreal+bysigma
    byminus = byappreal-bysigma

    bzplus = bzappreal+bzsigma
```

```
    bzminus = bzappreal-bzsigma

    plt.plot(time*1e6,bapplied*1e4,color='orange',label='bapplied')
    plt.plot(tspan*1e6,bxappreal*1e4,color='b',label='bx reconstructed')
    plt.fill_between(tspan*1e6,bxminus*1e4,bxplus*1e4,alpha = 0.5,color = 'b'
)
    plt.plot(tspan*1e6,byappreal*1e4,color='g',label='by reconstructed')
    plt.fill_between(tspan*1e6,byminus*1e4,byplus*1e4,alpha = 0.5,color = 'g'
)
    plt.plot(tspan*1e6,bzappreal*1e4,color='r',label='bz reconstructed')
    plt.fill_between(tspan*1e6,bzminus*1e4,bzplus*1e4,alpha = 0.5,color ='r')
    plt.legend()
    plt.xlabel('time (us)',fontsize=40)
    plt.legend(fontsize=20)
    plt.tick_params(labelsize=30)
    plt.axis([0,30,-.5,.5])


def fullreconstruct1(xshot,yshot,zshot,n):
    fig = plt.figure(facecolor='white')
    plt.subplot(1,3,1)
    plt.ylabel('B (Gauss)',fontsize=40)
    Breconstruct1(xshot+n,'x',xshot,yshot,zshot,n)

    plt.subplot(1,3,2)
    Breconstruct1(yshot+n,'y',xshot,yshot,zshot,n)

    plt.subplot(1,3,3)
    Breconstruct1(zshot+n,'z',xshot,yshot,zshot,n)



xshots = [11910,11943,11976,12009,12042,12075,12108,12141,12196,12241,12274,1
2307,12340,12373,12406,
            12447,12480,12513,12546,12579,12612,12656,12689,12733,12766,1280
4,12837,12870,12903,12936,
            12969,13002,13035,13068,13101,13134,13176,13210,13243,13276,1330
9,13353,13397,13430,13463]

yshots = [11921,11954,11987,12020,12053,12086,12119,12152,12207,12252,12285,1
2318,12351,12384,12417,
            12458,12491,12524,12557,12590,12634,12667,12700,12744,12777,1281
5,12848,12881,12914,12947,
            12980,13013,13046,13079,13112,13145,13187,13221,13254,13287,1333
1,13375,13408,13441,13474]
```

```
zshots = [11932,11965,11998,12031,12064,12097,12130,12185,12219,12263,12296,12329,12362,12395,12428,
             12469,12502,12535,12568,12601,12645,12678,12722,12755,12788,12826,12859,12892,12925,12958,
             12991,13024,13057,13090,13123,13165,13199,13232,13265,13298,13342,13386,13419,13452,13485]


def currentloop(x,y,z,I,a):
    offset = 0.03
    z=z+offset
    mu0=4*np.pi*10**-7

    rho=np.sqrt(x**2+y**2)
    r=np.sqrt(x**2+y**2+z**2)
    alpha = np.sqrt(a**2+r**2-2*a*rho)
    beta = np.sqrt(a**2+r**2+2*a*rho)
    k = np.sqrt(1-alpha**2/beta**2)
    gamma=x**2-y**2
    C=mu0*I/np.pi



    bx= (C*x*z)/(2*alpha**2*beta*rho**2)*((a**2+r**2)*special.ellipe(k**2)-alpha**2*special.ellipk(k**2))
    bx=np.nan_to_num(bx)

    by= (C*y*z)/(2*alpha**2*beta*rho**2)*((a**2+r**2)*special.ellipe(k**2)-alpha**2*special.ellipk(k**2))
    by=np.nan_to_num(by)

    bz= (C/(2*alpha**2*beta))*((a**2-r**2)*special.ellipe(k**2)+alpha**2*special.ellipk(k**2))
    np.nan_to_num(bz)



    return (bx,by,bz)
```

```
def biascoil(x,y,z,I,a):
    thickness=0.0038
    length = 0.04572
    width = 0.0342
    n=int(np.floor(length/thickness))
    w=int(np.floor(width/thickness))

    bx = 0
    by = 0
    bz = 0

    for i in range(0,n):
        for k in range (0,w):
            bx += currentloop(x,y,z+thickness*i,I,a+thickness*k)[0]
            by += currentloop(x,y,z+thickness*i,I,a+thickness*k)[1]
            bz += currentloop(x,y,z+thickness*i,I,a+thickness*k)[2]

    return (bx,by,bz)



"""FPGA channels associated to bdot probe chips"""
V_1_10_x ='\VDATAVALUES_RCARD2CH9'
T_1_10_x ='\VTIMEVALUES_RCARD2CH9'
V_1_10_y ='\VDATAVALUES_RCARD2CH5'
T_1_10_y ='\VTIMEVALUES_RCARD2CH5'
V_1_10_z ='\VDATAVALUES_RCARD2CH7'
T_1_10_z ='\VTIMEVALUES_RCARD2CH7'

V_1_9_x ='\VDATAVALUES_RCARD2CH3'
T_1_9_x ='\VTIMEVALUES_RCARD2CH3'
V_1_9_y ='\VDATAVALUES_RCARD2CH16'
T_1_9_y ='\VTIMEVALUES_RCARD2CH16'
V_1_9_z ='\VDATAVALUES_RCARD2CH1'
T_1_9_z ='\VTIMEVALUES_RCARD2CH1'

V_1_8_x ='\VDATAVALUES_RCARD2CH18'
T_1_8_x ='\VTIMEVALUES_RCARD2CH18'
V_1_8_y ='\VDATAVALUES_RCARD2CH22'
T_1_8_y ='\VTIMEVALUES_RCARD2CH22'
V_1_8_z ='\VDATAVALUES_RCARD2CH20'
T_1_8_z ='\VTIMEVALUES_RCARD2CH20'

V_2_1_x ='\VDATAVALUES_RCARD0CH0'
```

```
T_2_1_x ='\VTIMEVALUES_RCARD0CH0'
V_2_1_y ='\VDATAVALUES_RCARD0CH4'
T_2_1_y ='\VTIMEVALUES_RCARD0CH4'
V_2_1_z ='\VDATAVALUES_RCARD0CH2'
T_2_1_z ='\VTIMEVALUES_RCARD0CH2'

V_2_2_x ='\VDATAVALUES_RCARD0CH6'
T_2_2_x ='\VTIMEVALUES_RCARD0CH6'
V_2_2_y ='\VDATAVALUES_RCARD0CH10'
T_2_2_y ='\VTIMEVALUES_RCARD0CH10'
V_2_2_z ='\VDATAVALUES_RCARD0CH8'
T_2_2_z ='\VTIMEVALUES_RCARD0CH8'

V_2_3_x ='\VDATAVALUES_RCARD0CH12'
T_2_3_x ='\VTIMEVALUES_RCARD0CH12'
V_2_3_y ='\VDATAVALUES_RCARD0CH15'
T_2_3_y ='\VTIMEVALUES_RCARD0CH15'
V_2_3_z ='\VDATAVALUES_RCARD0CH14'
T_2_3_z ='\VTIMEVALUES_RCARD0CH14'

V_2_4_x ='\VDATAVALUES_RCARD0CH13'
T_2_4_x ='\VTIMEVALUES_RCARD0CH13'
V_2_4_y ='\VDATAVALUES_RCARD0CH9'
T_2_4_y ='\VTIMEVALUES_RCARD0CH9'
V_2_4_z ='\VDATAVALUES_RCARD0CH11'
T_2_4_z ='\VTIMEVALUES_RCARD0CH11'

V_2_5_x ='\VDATAVALUES_RCARD0CH7'
T_2_5_x ='\VTIMEVALUES_RCARD0CH7'
V_2_5_y ='\VDATAVALUES_RCARD0CH3'
T_2_5_y ='\VTIMEVALUES_RCARD0CH3'
V_2_5_z ='\VDATAVALUES_RCARD0CH5'
T_2_5_z ='\VTIMEVALUES_RCARD0CH5'

V_2_6_x ='\VDATAVALUES_RCARD0CH1'
T_2_6_x ='\VTIMEVALUES_RCARD0CH1'
V_2_6_y ='\VDATAVALUES_RCARD0CH18'
T_2_6_y ='\VTIMEVALUES_RCARD0CH18'
V_2_6_z ='\VDATAVALUES_RCARD0CH16'
T_2_6_z ='\VTIMEVALUES_RCARD0CH16'

V_2_7_x ='\VDATAVALUES_RCARD0CH20'
T_2_7_x ='\VTIMEVALUES_RCARD0CH20'
V_2_7_y ='\VDATAVALUES_RCARD0CH24'
```

```
T_2_7_y ='\VTIMEVALUES_RCARD0CH24'
V_2_7_z ='\VDATAVALUES_RCARD0CH22'
T_2_7_z ='\VTIMEVALUES_RCARD0CH22'

V_2_8_x ='\VDATAVALUES_RCARD0CH26'
T_2_8_x ='\VTIMEVALUES_RCARD0CH26'
V_2_8_y ='\VDATAVALUES_RCARD0CH30'
T_2_8_y ='\VTIMEVALUES_RCARD0CH30'
V_2_8_z ='\VDATAVALUES_RCARD0CH28'
T_2_8_z ='\VTIMEVALUES_RCARD0CH28'

V_2_9_x ='\VDATAVALUES_RCARD0CH31'
T_2_9_x ='\VTIMEVALUES_RCARD0CH31'
V_2_9_y ='\VDATAVALUES_RCARD0CH27'
T_2_9_y ='\VTIMEVALUES_RCARD0CH27'
V_2_9_z ='\VDATAVALUES_RCARD0CH29'
T_2_9_z ='\VTIMEVALUES_RCARD0CH29'

V_2_10_x ='\VDATAVALUES_RCARD0CH25'
T_2_10_x ='\VTIMEVALUES_RCARD0CH25'
V_2_10_y ='\VDATAVALUES_RCARD0CH21'
T_2_10_y ='\VTIMEVALUES_RCARD0CH21'
V_2_10_z ='\VDATAVALUES_RCARD0CH23'
T_2_10_z ='\VTIMEVALUES_RCARD0CH23'

V_2_11_x ='\VDATAVALUES_RCARD0CH19'
T_2_11_x ='\VTIMEVALUES_RCARD0CH19'
V_2_11_y ='\VDATAVALUES_RCARD1CH0'
T_2_11_y ='\VTIMEVALUES_RCARD1CH0'
V_2_11_z ='\VDATAVALUES_RCARD0CH17'
T_2_11_z ='\VTIMEVALUES_RCARD0CH17'

V_2_12_x ='\VDATAVALUES_RCARD1CH2'
T_2_12_x ='\VTIMEVALUES_RCARD1CH2'
V_2_12_y ='\VDATAVALUES_RCARD1CH6'
T_2_12_y ='\VTIMEVALUES_RCARD1CH6'
V_2_12_z ='\VDATAVALUES_RCARD1CH4'
T_2_12_z ='\VTIMEVALUES_RCARD1CH4'

V_2_13_x ='\VDATAVALUES_RCARD1CH8'
T_2_13_x ='\VTIMEVALUES_RCARD1CH8'
V_2_13_y ='\VDATAVALUES_RCARD1CH12'
T_2_13_y ='\VTIMEVALUES_RCARD1CH12'
V_2_13_z ='\VDATAVALUES_RCARD1CH10'
```

```
T_2_13_z ='\VTIMEVALUES_RCARD1CH10'

V_2_14_x ='\VDATAVALUES_RCARD1CH14'
T_2_14_x ='\VTIMEVALUES_RCARD1CH14'
V_2_14_y ='\VDATAVALUES_RCARD1CH13'
T_2_14_y ='\VTIMEVALUES_RCARD1CH13'
V_2_14_z ='\VDATAVALUES_RCARD1CH15'
T_2_14_z ='\VTIMEVALUES_RCARD1CH15'

V_2_15_x ='\VDATAVALUES_RCARD1CH11'
T_2_15_x ='\VTIMEVALUES_RCARD1CH11'
V_2_15_y ='\VDATAVALUES_RCARD1CH7'
T_2_15_y ='\VTIMEVALUES_RCARD1CH7'
V_2_15_z ='\VDATAVALUES_RCARD1CH9'
T_2_15_z ='\VTIMEVALUES_RCARD1CH9'

V_2_16_x ='\VDATAVALUES_RCARD1CH5'
T_2_16_x ='\VTIMEVALUES_RCARD1CH5'
V_2_16_y ='\VDATAVALUES_RCARD1CH1'
T_2_16_y ='\VTIMEVALUES_RCARD1CH1'
V_2_16_z ='\VDATAVALUES_RCARD1CH3'
T_2_16_z ='\VTIMEVALUES_RCARD1CH3'

V_2_17_x ='\VDATAVALUES_RCARD1CH16'
T_2_17_x ='\VTIMEVALUES_RCARD1CH16'
V_2_17_y ='\VDATAVALUES_RCARD1CH20'
T_2_17_y ='\VTIMEVALUES_RCARD1CH20'
V_2_17_z ='\VDATAVALUES_RCARD1CH18'
T_2_17_z ='\VTIMEVALUES_RCARD1CH18'

V_2_18_x ='\VDATAVALUES_RCARD1CH22'
T_2_18_x ='\VTIMEVALUES_RCARD1CH22'
V_2_18_y ='\VDATAVALUES_RCARD1CH26'
T_2_18_y ='\VTIMEVALUES_RCARD1CH26'
V_2_18_z ='\VDATAVALUES_RCARD1CH24'
T_2_18_z ='\VTIMEVALUES_RCARD1CH24'

V_2_19_x ='\VDATAVALUES_RCARD1CH28'
T_2_19_x ='\VTIMEVALUES_RCARD1CH28'
V_2_19_y ='\VDATAVALUES_RCARD1CH31'
T_2_19_y ='\VTIMEVALUES_RCARD1CH31'
V_2_19_z ='\VDATAVALUES_RCARD1CH30'
T_2_19_z ='\VTIMEVALUES_RCARD1CH30'
```

```
V_2_20_x ='\VDATAVALUES_RCARD1CH29'
T_2_20_x ='\VTIMEVALUES_RCARD1CH29'
V_2_20_y ='\VDATAVALUES_RCARD1CH25'
T_2_20_y ='\VTIMEVALUES_RCARD1CH25'
V_2_20_z ='\VDATAVALUES_RCARD1CH27'
T_2_20_z ='\VTIMEVALUES_RCARD1CH27'

V_2_21_x ='\VDATAVALUES_RCARD1CH23'
T_2_21_x ='\VTIMEVALUES_RCARD1CH23'
V_2_21_y ='\VDATAVALUES_RCARD1CH19'
T_2_21_y ='\VTIMEVALUES_RCARD1CH19'
V_2_21_z ='\VDATAVALUES_RCARD1CH21'
T_2_21_z ='\VTIMEVALUES_RCARD1CH21'

V_2_22_x ='\VDATAVALUES_RCARD1CH17'
T_2_22_x ='\VTIMEVALUES_RCARD1CH17'
V_2_22_y ='\VDATAVALUES_RCARD2CH2'
T_2_22_y ='\VTIMEVALUES_RCARD2CH2'
V_2_22_z ='\VDATAVALUES_RCARD2CH0'
T_2_22_z ='\VTIMEVALUES_RCARD2CH0'

V_2_23_x ='\VDATAVALUES_RCARD2CH4'
T_2_23_x ='\VTIMEVALUES_RCARD2CH4'
V_2_23_y ='\VDATAVALUES_RCARD2CH8'
T_2_23_y ='\VTIMEVALUES_RCARD2CH8'
V_2_23_z ='\VDATAVALUES_RCARD2CH6'
T_2_23_z ='\VTIMEVALUES_RCARD2CH6'

V_2_24_x ='\VDATAVALUES_RCARD2CH10'
T_2_24_x ='\VTIMEVALUES_RCARD2CH10'
V_2_24_y ='\VDATAVALUES_RCARD2CH14'
T_2_24_y ='\VTIMEVALUES_RCARD2CH14'
V_2_24_z ='\VDATAVALUES_RCARD2CH12'
T_2_24_z ='\VTIMEVALUES_RCARD2CH12'

V_2_25_x ='\VDATAVALUES_RCARD2CH15'
T_2_25_x ='\VTIMEVALUES_RCARD2CH15'
V_2_25_y ='\VDATAVALUES_RCARD2CH11'
T_2_25_y ='\VTIMEVALUES_RCARD2CH11'
V_2_25_z ='\VDATAVALUES_RCARD2CH13'
T_2_25_z ='\VTIMEVALUES_RCARD2CH13'

V_3_1_x ='\VDATAVALUES_RCARD2CH24'
T_3_1_x ='\VTIMEVALUES_RCARD2CH24'
```

```
V_3_1_y ='\VDATAVALUES_RCARD2CH28'
T_3_1_y ='\VTIMEVALUES_RCARD2CH28'
V_3_1_z ='\VDATAVALUES_RCARD2CH26'
T_3_1_z ='\VTIMEVALUES_RCARD2CH26'

V_3_2_x ='\VDATAVALUES_RCARD2CH30'
T_3_2_x ='\VTIMEVALUES_RCARD2CH30'
V_3_2_y ='\VDATAVALUES_RCARD2CH29'
T_3_2_y ='\VTIMEVALUES_RCARD2CH29'
V_3_2_z ='\VDATAVALUES_RCARD2CH31'
T_3_2_z ='\VTIMEVALUES_RCARD2CH31'

V_3_3_x ='\VDATAVALUES_RCARD2CH27'
T_3_3_x ='\VTIMEVALUES_RCARD2CH27'
V_3_3_y ='\VDATAVALUES_RCARD2CH23'
T_3_3_y ='\VTIMEVALUES_RCARD2CH23'
V_3_3_z ='\VDATAVALUES_RCARD2CH25'
T_3_3_z ='\VTIMEVALUES_RCARD2CH25'

V_3_4_x ='\VDATAVALUES_RCARD2CH21'
T_3_4_x ='\VTIMEVALUES_RCARD2CH21'
V_3_4_y ='\VDATAVALUES_RCARD2CH17'
T_3_4_y ='\VTIMEVALUES_RCARD2CH17'
V_3_4_z ='\VDATAVALUES_RCARD2CH19'
T_3_4_z ='\VTIMEVALUES_RCARD2CH19'

def Bfield(shotnum,attenuation,correctionangle):
    angle=correctionangle*np.pi/180

    def rotmatrix(theta):
        return np.array([[np.cos(theta),-1*np.sin(theta),0],[np.sin(theta),np
.cos(theta),0],[0,0,1]])

    tree = mds.Tree('op_tree', shotnum)
    tnode = tree.getNode(T_2_2_x)
    mds_time = tnode.getData()
    time = np.array(mds_time)

    fs=50e6
    lowcut=13000
    highcut=23e6


    xcalshotlist=[12141,12196,12241,12274,12307,12340,12373,12406,
```

```
            12447,12480,12513,12546,12579,12612,12656,12689,12733,12766,1280
4,12837,12870,12903,12936,
            12969,13002,13035,13068,13101,13134,13176,13210,13243][::-1]

    ycalshotlist=[12152,12207,12252,12285,12318,12351,12384,12417,
            12458,12491,12524,12557,12590,12634,12667,12700,12744,12777,1281
5,12848,12881,12914,12947,
            12980,13013,13046,13079,13112,13145,13187,13221,13254][::-1]

    zcalshotlist=[12185,12219,12263,12296,12329,12362,12395,12428,
            12469,12502,12535,12568,12601,12645,12678,12722,12755,12788,1282
6,12859,12892,12925,12958,
            12991,13024,13057,13090,13123,13165,13199,13232,13265][::-1]


    vxcardlist=[V_3_4_x,V_3_3_x,V_3_2_x,V_3_1_x,V_2_25_x,V_2_24_x,V_2_23_x,V_
2_22_x,V_2_21_x,V_2_20_x,
              V_2_19_x,V_2_18_x,V_2_17_x,V_2_16_x,V_2_15_x,V_2_14_x,V_2_13_x
,V_2_12_x,V_2_11_x,
               V_2_10_x,V_2_9_x,V_2_8_x,V_2_7_x,V_2_6_x,V_2_5_x,V_2_4_x,V_2_
3_x,V_2_2_x,V_2_1_x,
               V_1_10_x,V_1_9_x,V_1_8_x]

    vycardlist=[V_3_4_y,V_3_3_y,V_3_2_y,V_3_1_y,V_2_25_y,V_2_24_y,V_2_23_y,V_
2_22_y,V_2_21_y,V_2_20_y,
              V_2_19_y,V_2_18_y,V_2_17_y,V_2_16_y,V_2_15_y,V_2_14_y,V_2_13_y
,V_2_12_y,V_2_11_y,
               V_2_10_y,V_2_9_y,V_2_8_y,V_2_7_y,V_2_6_y,V_2_5_y,V_2_4_y,V_2_
3_y,V_2_2_y,V_2_1_y,
               V_1_10_y,V_1_9_y,V_1_8_y]

    vzcardlist=[V_3_4_z,V_3_3_z,V_3_2_z,V_3_1_z,V_2_25_z,V_2_24_z,V_2_23_z,V_
2_22_z,V_2_21_z,V_2_20_z,
              V_2_19_z,V_2_18_z,V_2_17_z,V_2_16_z,V_2_15_z,V_2_14_z,V_2_13_z
,V_2_12_z,V_2_11_z,
               V_2_10_z,V_2_9_z,V_2_8_z,V_2_7_z,V_2_6_z,V_2_5_z,V_2_4_z,V_2_
3_z,V_2_2_z,V_2_1_z,
               V_1_10_z,V_1_9_z,V_1_8_z]

    xcancel =[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1
]
    ycancel =[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
]
```

```
    zcancel =[1,1,1,1,1,1,1,0,1,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,0
]


    xflip =[1,1,1,1,1,1,1,-1,-1,1,1,-1,-1,1,1,1,1,1,-1,1,1,1,1,-1,1,1,1,1,1,1
,1,1]
    yflip =[1,1,1,1,1,1,1,1,1,1,1,1,-1,1,1,1,1,1,1,-1,1,1,1,-1,1,1,1,1,1,1,1,
1]
    zflip =[1,1,1,1,1,1,1,4.5,1,1,1,1,-1,1,1,1,1,1,-1,-1,1,1,1,3.6,1,1,1,1,1,
1,1,1]


    rawvxlist = [[] for _ in range(32)]
    rawvylist = [[] for _ in range(32)]
    rawvzlist = [[] for _ in range(32)]

    vxlist = [[] for _ in range(32)]
    vylist = [[] for _ in range(32)]
    vzlist = [[] for _ in range(32)]


    integrated_vxlist=[[] for _ in range(32)]
    integrated_vylist=[[] for _ in range(32)]
    integrated_vzlist=[[] for _ in range(32)]


    bx=[[] for _ in range(32)]
    by=[[] for _ in range(32)]
    bz=[[] for _ in range(32)]

    bxsigma=[[] for _ in range(32)]
    bysigma=[[] for _ in range(32)]
    bzsigma=[[] for _ in range(32)]

    bxplus=[[] for _ in range(32)]
    byplus=[[] for _ in range(32)]
    bzplus=[[] for _ in range(32)]

    bxminus=[[] for _ in range(32)]
    byminus=[[] for _ in range(32)]
    bzminus=[[] for _ in range(32)]

    b=[[] for _ in range(32)]
    bplus=[[] for _ in range(32)]
```

```
    bminus=[[] for _ in range(32)]

    for i in range(0,32):
        m = fulltest(xcalshotlist[i],ycalshotlist[i],zcalshotlist[i],4)[0]
        msigma= fulltest(xcalshotlist[i],ycalshotlist[i],zcalshotlist[i],4)[1
]

        vxnode=tree.getNode(vxcardlist[i])
        vynode=tree.getNode(vycardlist[i])
        vznode=tree.getNode(vzcardlist[i])
        mds_vx=vxnode.getData()
        mds_vy=vynode.getData()
        mds_vz=vznode.getData()

        rawvx=np.array(mds_vx)
        rawvy=np.array(mds_vy)
        rawvz=np.array(mds_vz)

        tstart=np.where(time*1e6>6)[0][0]
        tend=np.where(time*1e6>200)[0][0]
        tspan=time[:tend]

        offsetx=np.mean(rawvx[:tstart])
        offsety=np.mean(rawvy[:tstart])
        offsetz=np.mean(rawvz[:tstart])


        vx=(rawvx-offsetx)*attenuation*xcancel[i]
        vy=(rawvy-offsety)*attenuation*ycancel[i]
        vz=(rawvz-offsetz)*attenuation*zcancel[i]

        vxfilter=butter_bandpass_filter(vx,lowcut,highcut,fs,order=3)
        vyfilter=butter_bandpass_filter(vy,lowcut,highcut,fs,order=3)
        vzfilter=butter_bandpass_filter(vz,lowcut,highcut,fs,order=3)


        vxint=integrate.cumtrapz(vxfilter[:tend],tspan,initial=0)
        vyint=integrate.cumtrapz(vyfilter[:tend],tspan,initial=0)
        vzint=integrate.cumtrapz(vzfilter[:tend],tspan,initial=0)

        rawvxlist[i]=vx
        rawvylist[i]=vy
        rawvzlist[i]=vz
```

```
        vxlist[i]=vxfilter
        vylist[i]=vyfilter
        vzlist[i]=vzfilter


        integrated_vxlist[i]=vxint
        integrated_vylist[i]=vyint
        integrated_vzlist[i]=vzint


        bx[i] = (m[0][0]*vxint*xflip[i] + m[0][1]*vyint*yflip[i] + m[0][2]*vz
int*zflip[i])
        by[i] = (m[1][0]*vxint*xflip[i] + m[1][1]*vyint*yflip[i] + m[1][2]*vz
int*zflip[i])
        bz[i] = (m[2][0]*vxint*xflip[i] + m[2][1]*vyint*yflip[i] + m[2][2]*vz
int*zflip[i])

        bxsigma[i] = np.sqrt((msigma[0][0]*vxint)**2+(msigma[0][1]*vyint)**2+
(msigma[0][2]*vzint)**2)
        bysigma[i] = np.sqrt((msigma[1][0]*vxint)**2+(msigma[1][1]*vyint)**2+
(msigma[1][2]*vzint)**2)
        bzsigma[i] = np.sqrt((msigma[2][0]*vxint)**2+(msigma[2][1]*vyint)**2+
(msigma[2][2]*vzint)**2)

        bxplus[i] = bx[i]+bxsigma[i]
        byplus[i] = by[i]+bysigma[i]
        bzplus[i] = bz[i]+bzsigma[i]

        bxminus[i] = bx[i]-bxsigma[i]
        byminus[i] = by[i]-bysigma[i]
        bzminus[i] = bz[i]-bzsigma[i]

        b[i]=np.dot((rotmatrix(angle)),[bx[i],by[i],bz[i]])
        bplus[i]=np.dot((rotmatrix(angle)),[bxplus[i],byplus[i],bzplus[i]])
        bminus[i]=np.dot((rotmatrix(angle)),[bxminus[i],byminus[i],bzminus[i]
])

    b[20][2][:]=b[19][2][:]*0
    return (b,bplus,bminus,tspan)
```

In [21]:

```
# chiporder=['near.4','near.3','near.2','near.1','center.25','center.24','cen
ter.23','center.22','center.21','center.20','center.19','center.18',
```

```
#            'center.17','center.16','center.15','center.14','center.13','cen
ter.12','center.11','center.10','center.9','center.8','center.7',
#            'center.6','center.5','center.4','center.3','center.2','center.1
','far.10','far.9','far.8']

chiporder=['3.4','3.3','3.2','3.1','2.25','2.24','2.23','2.22','2.21','2.20',
'2.19','2.18',
           '2.17','2.16','2.15','2.14','2.13','2.12','2.11','2.10','2.9','2.
8','2.7',
           '2.6','2.5','2.4','2.3','2.2','2.1','1.10','1.9','1.8']
```

```
C:\Users\Sander\Anaconda2\lib\site-packages\matplotlib\collections.py:590: Fu
tureWarning: elementwise comparison failed; returning scalar instead, but in
the future will perform elementwise comparison
  if self._edgecolors == str('face'):
```

```
data = Bfield(6999,100,5)
B = data[0]
Bplus = data[1]
Bminus = data[2]
time = data[3]


def lineplots(b,bplus,bminus,timedata,time):
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe

    chippositions = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -
7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])

    timeindex = np.where(timedata*1e6>time)[0][0]

    t = timedata[timeindex]*1e6

    Barray = np.array(b)
    Bplusarray = np.array(bplus)
```

```
    Bminusarray= np.array(bminus)

    Bx =np.delete(Barray[:,2,timeindex][::-1],(0,3,8,10,11,22,24),0) + biasco
il(chippositionsx/100,0,zprobe,I,a)[0]
    By = np.delete(Barray[:,1,timeindex][::-1],3,0)+ biascoil(chippositions/1
00,0,zprobe,I,a)[1]
    Bz = np.delete(Barray[:,0,timeindex][::-1],3,0)+ biascoil(chippositions/1
00,0,zprobe,I,a)[2]

    bxerror = np.delete(Bplusarray[:,2,timeindex][::-1]-Bminusarray[:,2,timei
ndex][::-1],(0,3,8,10,11,22,24),0)/2
    byerror = np.delete(Bplusarray[:,1,timeindex][::-1]-Bminusarray[:,1,timei
ndex][::-1],3,0)/2
    bzerror = np.delete(Bplusarray[:,0,timeindex][::-1]-Bminusarray[:,0,timei
ndex][::-1],3,0)/2


    Byp = np.delete(Barray[:,1,timeindex][::-1],(0,3,8,10,11,22,24),0) + bias
coil(chippositionsx/100,0,zprobe,-I,a)[1]
    Bzp = np.delete(Barray[:,0,timeindex][::-1],(0,3,8,10,11,22,24),0) + bias
coil(chippositionsx/100,0,zprobe,-I,a)[2]

#    fig = plt.figure(facecolor='white',figsize = (17,13))
    plt.errorbar(chippositionsx,Bx, yerr=bxerror,marker ='.',markersize=20, c
olor = 'b',linewidth =5,elinewidth=3,label = 'Bx')
    plt.errorbar(chippositions,By, yerr=byerror,marker ='.',markersize=20, co
lor = 'g',linewidth =5,elinewidth=3, label = 'By')
    plt.errorbar(chippositions,Bz, yerr=bzerror,marker ='.',markersize=20, co
lor = 'r',linewidth =5,elinewidth=3, label = 'Bz')
#    plt.plot(chippositionsx,2*np.pi*abs(chippositionsx/100)*Bzp/(1.08*np.sqr
t(Byp**2+Bx**2)))
    plt.axis([-30,30,-0.2,0.2])
#    plt.xlabel('position (cm)',fontsize=20)
#    plt.ylabel('B (Tesla)',fontsize=20)
#    plt.legend(fontsize=20)
    plt.grid(True)
    plt.title(r't = %d $\mu$ s' %t, fontsize = 45,loc = 'right')
    plt.tick_params(labelsize=30)
```

```
def lineplots2(b,bplus,bminus,timedata,time):
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe

    chippositions = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -
7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])

    timeindex = np.where(timedata*1e6>time)[0][0]

    t = timedata[timeindex]*1e6

    Barray = np.array(b)
    Bplusarray = np.array(bplus)
    Bminusarray= np.array(bminus)

    Bx =np.delete(Barray[:,2,timeindex][::-1],(0,3,8,10,11,22,24),0) + biasco
il(chippositionsx/100,0,zprobe,I,a)[0]
    By = np.delete(Barray[:,1,timeindex][::-1],3,0)+ biascoil(chippositions/1
00,0,zprobe,I,a)[1]
    Bz = np.delete(Barray[:,0,timeindex][::-1],3,0)+ biascoil(chippositions/1
00,0,zprobe,I,a)[2]

    bxplus= np.delete(Bplusarray[:,2,timeindex][::-1],(0,3,8,10,11,22,24),0)
+ biascoil(chippositionsx/100,0,zprobe,I,a)[0]
    byplus = np.delete(Bplusarray[:,1,timeindex][::-1],3,0)+ biascoil(chippos
itions/100,0,zprobe,I,a)[1]
    bzplus =  np.delete(Bplusarray[:,0,timeindex][::-1],3,0)+ biascoil(chippo
sitions/100,0,zprobe,I,a)[2]

    bxminus= np.delete(Bminusarray[:,2,timeindex][::-1],(0,3,8,10,11,22,24),0
) + biascoil(chippositionsx/100,0,zprobe,I,a)[0]
    byminus = np.delete(Bminusarray[:,1,timeindex][::-1],3,0)+ biascoil(chipp
ositions/100,0,zprobe,I,a)[1]
    bzminus =  np.delete(Bminusarray[:,0,timeindex][::-1],3,0)+ biascoil(chip
positions/100,0,zprobe,I,a)[2]
```

```
#    fig = plt.figure(facecolor='white',figsize = (17,13))
    plt.plot(chippositionsx,Bx, marker ='.',markersize=20, color = 'b',linewi
dth =5,label = 'Bx')
    plt.fill_between(chippositionsx,bxminus,bxplus,alpha = 0.5,color = 'b')

    plt.plot(chippositions,By, marker ='.',markersize=20, color = 'g',linewid
th =5, label = 'By')
    plt.fill_between(chippositions,byminus,byplus,alpha = 0.5,color = 'g')

    plt.plot(chippositions,Bz, marker ='.',markersize=20, color = 'r',linewid
th =5, label = 'Bz')
    plt.fill_between(chippositions,bzminus,bzplus,alpha = 0.5,color = 'r')

    plt.axis([-30,30,-0.5,0.5])
    plt.xlabel('position (cm)',fontsize=40)
    plt.ylabel('B (Tesla)',fontsize=40)
    plt.legend(fontsize=40)
    plt.grid(True)
    plt.title(r't = %d $\mu$ s' %t, fontsize = 45,loc = 'right')
    plt.tick_params(labelsize=40)


def contourplots(Bdata,timedata):
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe

    chippositions = np.array([-24,-20.4,-16.8,-13.2, -12.1, -11. ,  -9.9,  -8
.8,  -7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])

    chippositionsYZ = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,
-7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
```

```
          6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)


    X, Y = np.meshgrid(timedata*1e6,chippositionsYZ)

    Xp,Yp = np.meshgrid(timedata*1e6,chippositionsx)

    biasx=biascoil(chippositions/100,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions/100,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions/100,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx =np.delete(Barray[:,2,:][::-1]+biasxlist,(0,3,8,10,11,22,24),0)
    By = np.delete(Barray[:,1,:][::-1]+ biasylist,3,0)
    Bz = np.delete(Barray[:,0,:][::-1]+ biaszlist,3,0)
    Bmag = np.delete(np.sqrt((Barray[:,2,:][::-1]+biasxlist)**2+(Barray[:,:
][::-1])**2+(Barray[:,0,:][::-1])**2),(0,3,8,10,11,22,24),0)
    Byp =np.delete(Barray[:,1,:][::-1]+biasylist,(0,3,8,10,11,22,24),0)


    fig1 = plt.figure(facecolor='white',figsize = (17,13))
    plt.subplot(3,1,1)
    levels = np.arange(-0.5,.5,0.002)
    cf=plt.contourf(Xp,Yp,Bx,levels=levels)
    plt.contour(Xp,Yp,Bx,colors = 'k')
    plt.axis([0,120,-20.4,27.6])
    plt.tick_params(labelsize=30)
    plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom='on',      # ticks along the bottom edge are off
    top='on',         # ticks along the top edge are off
    labelbottom='off') # labels along the bottom edge are off


    plt.subplot(3,1,2)
    plt.contourf(X,Y,By,levels=levels)
    plt.contour(X,Y,By,colors = 'k')
```

```
    plt.axis([0,120,-20.4,27.6])
    plt.tick_params(labelsize=30)
    plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom='on',      # ticks along the bottom edge are off
    top='on',         # ticks along the top edge are off
    labelbottom='off') # labels along the bottom edge are off


    plt.subplot(3,1,3)
    plt.contourf(X,Y,Bz,levels=levels)
    plt.contour(X,Y,Bz,colors='k')
#    cbar = fig1.colorbar(cf)
#    cbar.ax.tick_params(labelsize=20)
    plt.axis([0,120,-20.4,27.6])
    plt.tick_params(labelsize=30)

def crossection(Bdata,timedata):
    mu0=4*np.pi*10**-7
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe

    chippositions = np.array([-24,-20.4,-16.8,-13.2, -12.1, -11. ,  -9.9,  -8
.8,  -7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])

    chippositionsYZ = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,
-7.7,  -6.6,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6]
)


    X, Y = np.meshgrid(timedata*1e6,chippositionsYZ)
```

```
    Xp,Yp = np.meshgrid(timedata*1e6,chippositionsx)

    biasx=biascoil(chippositions/100,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions/100,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions/100,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx =np.delete(Barray[:,2,:][::-1]+biasxlist,(0,3,8,10,11,22,24),0)
    By = np.delete(Barray[:,1,:][::-1]+ biasylist,3,0)
    Bz = np.delete(Barray[:,0,:][::-1]+ biaszlist,3,0)
    Bthetamag = np.delete(np.sqrt((Barray[:,2,:][::-1]+biasxlist)**2+(Barray[
:,1,:][::-1]+ biasylist)**2),(0,3,8,10,11,22,24),0)
    Byp =np.delete(Barray[:,1,:][::-1]+biasylist,(0,3,8,10,11,22,24),0)

#setup grid for interpolation for streamlines
    y = timedata*1e6
    x =np.linspace(-13.2,13.2,100)
    xi,yi = np.meshgrid(x,y)

#interpolate data onto grid
    pxx = Xp.flatten()
    pyx = Yp.flatten()

    px=X.flatten()
    py=Y.flatten()

    pbx =Bx.flatten()
    pby = By.flatten()


    gbx = griddata(zip(pxx,pyx),pbx,(yi,xi),method='linear')
    gby = griddata(zip(px,py),pby,(yi,xi),method='linear')


    fig1=plt.figure(facecolor='white',figsize = (13,17))
    plt.ylabel('time [us]',fontsize=30)
    plt.xlabel('X position [cm]',fontsize=30)
    levels = np.arange(-0.5,.5,0.002)
    cf = plt.contourf(Y,X,Bz,levels=levels)
```

```
    plt.colorbar(cf)
#    plt.contour(Yp,Xp,Bthetamag,colors = 'k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Yp[skip],Xp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwidt
h = 3,scale= .3,scale_units = 'inches')
    lw = 10*np.sqrt(gbx**2+gby**2)/np.sqrt(gbx**2+gby**2).max()
    plt.streamplot(x,y,gbx,-1*gby,density = 4, color = 'r',linewidth = lw)
    plt.tick_params(labelsize=30)

#    plt.axis([-14,14,80,120])

    fig2=plt.figure(facecolor='white',figsize = (13,17))
    plt.ylabel('time [us]',fontsize=30)
    plt.xlabel('X position [cm]',fontsize=30)
    levels = np.arange(-0.5,.5,0.002)
    cf = plt.contourf(Y,-X,Bz,levels=levels)
    plt.colorbar(cf)
#    plt.contour(Yp,Xp,Bthetamag,colors = 'k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Yp[skip],-Xp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwid
th = 3,scale= .3,scale_units = 'inches')
    lw = 10*np.sqrt(gbx**2+gby**2)/np.sqrt(gbx**2+gby**2).max()
    plt.streamplot(x,-y,gbx,-1*gby,density = 4, color = 'r',linewidth = lw)
    plt.tick_params(labelsize=30)
#    plt.axis([-14,14,80,120])


def current(Bdata,timedata,tstart,tend,span):
    mu0=4*np.pi*10**-7
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe
    scaley = span/(tend-tstart)

    chippositions = np.array([-24,-20.4,-16.8,-13.2, -12.1, -11. ,  -9.9,  -8
.8,  -7.7,  -6.6,  -5.5,  -4.4,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/10
0

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/100
```

```
    chippositionsYZ = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,
-7.7,  -6.6,  -5.5,  -4.4,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/10
0


    tstartindex = np.where(timedata*1e6>tstart)[0][0]
    tendindex = np.where(timedata*1e6>tend)[0][0]

    shift = -((timedata[tendindex]*1e6+timedata[tstartindex]*1e6)/2)*scaley


    Y, X = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chipp
ositionsYZ)

    Yp,Xp = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chip
positionsx)

    biasx=biascoil(chippositions,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx =np.delete(Barray[:,2,tstartindex:tendindex][::-1]+biasxlist,(0,3,8,10
,11,22,24),0)
    By = np.delete(Barray[:,1,tstartindex:tendindex][::-1]+ biasylist,3,0)
    Bz = np.delete(Barray[:,0,tstartindex:tendindex][::-1]+ biaszlist,3,0)
    Byp =np.delete(Barray[:,1,tstartindex:tendindex][::-1]+biasylist,(0,3,8,1
0,11,22,24),0)
    Bzp = np.delete(Barray[:,0,tstartindex:tendindex][::-1]+biaszlist,(0,3,8,
10,11,22,24),0)

    #note that first term [0] is gradient in rows (i.e. x position) and secon
d term is gradient in columns (i.e. time)
    #since y points downward, this first gradient in bx is for time descendin
g alon y axis, corresponding to probe transit top to bottom
```

```
    gradientbx=np.gradient(Bx,chippositionsx,timedata[tstartindex:tendindex]*
1e6*scaley,edge_order=2)
    gradientby=np.gradient(Byp,chippositionsx,timedata[tstartindex:tendindex]
*1e6*scaley,edge_order=2)




    fig1 = plt.figure(facecolor='white',figsize=(10,10))
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,1e6)
    cf = plt.contourf(Xp,Yp,1/mu0*(gradientby[0]+gradientbx[1]),levels=levels
)#add grad bx since db/dt=-db/dy
    plt.colorbar(cf)
    plt.contour(X,Y,Bz,20,colors='k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Xp[skip],Yp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwidt
h = 3,scale= .5,scale_units = 'inches')
    plt.axis([.02,.11,-span/2,span/2])

    plt.subplot(2,1,2)
    levels2=np.array([-8,-6,-3.11,-2,0,8])
    cf2 = plt.contourf(Xp,Yp,(gradientby[0]+gradientbx[1])/Bzp*(span),levels=
levels2)
    plt.colorbar(cf2)
    plt.contour(X,Y,Bz,20,colors='k')
    plt.axis([.02,.11,-span/2,span/2])



    fig2 = plt.figure(facecolor='white',figsize=(10,10))
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,1e6)
    cf = plt.contourf(Xp,-Yp,1/mu0*(gradientby[0]-gradientbx[1]),levels=level
s)
    plt.colorbar(cf)
    plt.contour(X,-Y,Bz,20,colors='k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Xp[skip],-Yp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwid
th = 3,scale= .5,scale_units = 'inches')
    plt.axis([.02,.11,-span/2,span/2])

    plt.subplot(2,1,2)
    levels2=np.array([-8,-6,-3.11,-2,0,8])
```

```
    cf2 = plt.contourf(Xp,-Yp,(gradientby[0]-gradientbx[1])/Bzp*(span),levels
=levels2)
    plt.colorbar(cf2)
    plt.contour(X,-Y,Bz,20,colors='k')
    plt.axis([.02,.11,-span/2,span/2])


def currentinterp(Bdata,timedata,tstart,tend,span,radius):
    mu0=4*np.pi*10**-7
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .21 #axial location of bdot probe
    scaley = span/(tend-tstart)

    chippositions = np.array([-24,-20.4,-16.8,-13.2, -12.1, -11. ,  -9.9,  -8
.8,  -7.7,  -6.6,  -5.5,  -4.4,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/10
0

    chippositionsx = np.array([-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,  -6.6
,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6, 8.8,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/100

    chippositionsYZ = np.array([-24,-20.4,-16.8, -12.1, -11. ,  -9.9,  -8.8,
-7.7,  -6.6,  -5.5,  -4.4,
    -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
     6.6,   7.7,   8.8,   9.9,  11. ,  12.1,  13.2, 16.8, 20.4, 24, 27.6])/10
0


    tstartindex = np.where(timedata*1e6>tstart)[0][0]
    tendindex = np.where(timedata*1e6>tend)[0][0]

    shift = -((timedata[tendindex]*1e6+timedata[tstartindex]*1e6)/2)*scaley


    Y, X = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chipp
ositionsYZ)

    Yp,Xp = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chip
positionsx)
```

```
    biasx=biascoil(chippositions,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx =np.delete(Barray[:,2,tstartindex:tendindex][::-1]+biasxlist,(0,3,8,10
,11,22,24),0)
    By = np.delete(Barray[:,1,tstartindex:tendindex][::-1]+ biasylist,3,0)
    Bz = np.delete(Barray[:,0,tstartindex:tendindex][::-1]+ biaszlist,3,0)
    Byp =np.delete(Barray[:,1,tstartindex:tendindex][::-1]+biasylist,(0,3,8,1
0,11,22,24),0)
    Bzp =np.delete(Barray[:,0,tstartindex:tendindex][::-1]+biaszlist,(0,3,8,1
0,11,22,24),0)

    #setup grid for interpolated values
    npts=100
    #y=timedata[tstartindex:tendindex]*1e6*scaley+shift
    y=np.linspace(timedata[tstartindex]*1e6*scaley+shift,timedata[tendindex]*
1e6*scaley+shift,npts)
    x =np.linspace(-13.2,13.2,npts)/100
    xi,yi = np.meshgrid(x,y)

    #interpolate data onto grid
    pxx=Xp.flatten() #Bx data points
    pyx=Yp.flatten() #Bx data points

    px=X.flatten() #by,bz data points
    py=Y.flatten() #by,bz data points

    pbx =Bx.flatten()
    pby = By.flatten()
    pbz = Bz .flatten()

    gbx = griddata(zip(pxx,pyx),pbx,(xi,yi),method='linear')
    gby= griddata(zip(px,py),pby,(xi,yi),method='linear')
    gbz = griddata(zip(px,py),pbz,(xi,yi),method='linear')

    #note that first term [0] is gradient in rows (now time) and second term
is gradient in columns (now chip position)
```

```
    #not sure why the order changed from the previous current command
    #since y points downward, this first gradient in bx is for time descendin
g alon y axis, corresponding to probe transit top to bottom
    gradientbx=np.gradient(gbx,y,x,edge_order=2)
    gradientby=np.gradient(gby,y,x,edge_order=2)


    fig1 = plt.figure(facecolor='white',figsize=(12,14))
    plt.subplot(2,1,1)
    levels=np.arange(-5e7,5e7,1e6)
    cf = plt.contourf(xi,yi,1/mu0*(gradientby[1]+gradientbx[0]),levels=levels
,cmap='bwr')#add grad bx since db/dt=-db/dy
    cb=plt.colorbar(cf)
    plt.contour(xi,yi,gbz,20,colors='k',linewidths=2)
    skip = (slice(None,None,5),slice(None,None,None))
    plt.quiver(xi[skip],yi[skip],gbx[skip],-1*gby[skip],width = 0.003,headwid
th = 3,scale= .5,scale_units = 'inches')
    plt.axis([0.01,.11,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb.ax.tick_params(labelsize=25)

    plt.subplot(2,1,2)
#    levels2=np.array([-20,-10,-3.11,0,20])
    levels2 = np.arange(-200,20,20)
    cf2 = plt.contourf(xi,yi,(gradientby[1]+gradientbx[0])/gbz*radius,levels=
levels2,cmap='Blues_r')
    cb2=plt.colorbar(cf2)
    plt.contour(xi,yi,gbz,20,colors='k',linewidths=2)
    plt.axis([0.01,.11,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb2.ax.tick_params(labelsize=25)
    plt.tight_layout()

    fig2 = plt.figure(facecolor='white',figsize=(12,14))
    plt.subplot(2,1,1)
    levels=np.arange(-1e7,1e7,1e6)
    cf = plt.contourf(xi,-yi,1/mu0*(gradientby[1]-gradientbx[0]),levels=level
s,cmap='bwr')
    cb=plt.colorbar(cf)
    plt.contour(xi,-yi,gbz,20,colors='k',linewidths=2)
```

```
    skip = (slice(None,None,5),slice(None,None,None))
    plt.quiver(xi[skip],-yi[skip],gbx[skip],-1*gby[skip],width = 0.003,headwi
dth = 3,scale= .5,scale_units = 'inches')
    plt.axis([0.01,.11,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb.ax.tick_params(labelsize=25)


    plt.subplot(2,1,2)
#    levels2=np.array([-20,-10,-3.11,0,20])
    levels2 = np.arange(-100,20,20)
    cf2 = plt.contourf(xi,-yi,(gradientby[1]-gradientbx[0])/gbz*radius,levels
=levels2,cmap='Blues_r')
    cb2=plt.colorbar(cf2)
    plt.contour(xi,-yi,gbz,20,colors='k',linewidths=2)
    plt.axis([0.01,.11,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb2.ax.tick_params(labelsize=25)
    plt.tight_layout()
```

The following script is for magnetic probe array in the third operational campaign when all 3 probes have been mounted.

```
import MDSplus as mds
import numpy as np
from numpy.linalg import inv
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from scipy.optimize import minimize
import scipy.fftpack as fft
from scipy.fftpack import rfft,irfft,fftfreq
import scipy.integrate as integrate
from math import factorial
from scipy.signal import butter, lfilter
from scipy.signal import freqz
from scipy import special
from scipy.interpolate import griddata
from scipy import stats
%matplotlib
Using matplotlib backend: Qt4Agg

def multerror(val1,val2,sigma1,sigma2):
    sigma = abs(val1*val2)*np.sqrt((sigma1/val1)**2+(sigma2/val2)**2)
    return sigma

def addsuberror(sigma1,sigma2):
    sigma = np.sqrt(sigma1**2+sigma2**2)
    return sigma

def det(a,b,c,d,sa,sb,sc,sd):
    ans = a*d-b*c
    e1 = multerror(a,d,sa,sd)
    e2 = multerror(b,c,sb,sc)
    error = addsuberror(e1,e2)
    return ans,error


def matrixinverse(matrix,sigmamatrix):
    realdet = np.linalg.det(matrix)

    a = matrix[0][0]
    b = matrix[0][1]
    c = matrix[0][2]
```

```
    d = matrix[1][0]
    e = matrix[1][1]
    f = matrix[1][2]
    g = matrix[2][0]
    h = matrix[2][1]
    k = matrix[2][2]

    aa = sigmamatrix[0][0]
    bb = sigmamatrix[0][1]
    cc = sigmamatrix[0][2]
    dd = sigmamatrix[1][0]
    ee = sigmamatrix[1][1]
    ff = sigmamatrix[1][2]
    gg = sigmamatrix[2][0]
    hh = sigmamatrix[2][1]
    kk = sigmamatrix[2][2]


    determinant = a*det(e,f,h,k,ee,ff,hh,kk)[0] - b*det(d,f,g,k,dd,ff,gg,kk)[
0] + c*det(d,e,g,h,dd,ee,gg,hh)[0]

    e1 = multerror(a,det(e,f,h,k,ee,ff,hh,kk)[0],aa,det(e,f,h,k,ee,ff,hh,kk)[
1])
    e2 = multerror(b,det(d,f,g,k,dd,ff,gg,kk)[0],bb,det(d,f,g,k,dd,ff,gg,kk)[
1])
    e3 = multerror(c,det(d,e,g,h,dd,ee,gg,hh)[0],cc,det(d,e,g,h,dd,ee,gg,hh)[
1])

    deterror = np.sqrt(e1**2+e2**2+e3**2)

    detpererror = deterror/determinant

    minv = np.array([[1/determinant*det(e,f,h,k,ee,ff,hh,kk)[0],1/determinant
*det(c,b,k,h,cc,bb,kk,hh)[0],1/determinant*det(b,c,e,f,bb,cc,ee,ff)[0]],
                            [1/determinant*det(f,d,k,g,ff,dd,kk,gg)[0],1/determ
inant*det(a,c,g,k,aa,cc,gg,kk)[0],1/determinant*det(c,a,f,d,cc,aa,ff,dd)[0]],
                            [1/determinant*det(d,e,g,h,dd,ee,gg,hh)[0],1/determ
inant*det(b,a,h,g,bb,aa,hh,gg)[0],1/determinant*det(a,b,d,e,aa,bb,dd,ee)[0]]]
)

    sigmainv = np.array([[multerror(1/determinant,det(e,f,h,k,ee,ff,hh,kk)[0]
,1/determinant*detpererror,det(e,f,h,k,ee,ff,hh,kk)[1]),
                  multerror(1/determinant,det(c,b,k,h,cc,bb,kk,hh)[0],1/determ
inant*detpererror,det(c,b,k,h,cc,bb,kk,hh)[1]),
```

```
                    multerror(1/determinant,det(c,b,k,h,cc,bb,kk,hh)[0],1/determ
inant*detpererror,det(c,b,k,h,cc,bb,kk,hh)[1])],
                   [multerror(1/determinant,det(f,d,k,g,ff,dd,kk,gg)[0],1/determ
inant*detpererror,det(f,d,k,g,ff,dd,kk,gg)[1]),
                    multerror(1/determinant,det(a,c,g,k,aa,cc,gg,kk)[0],1/determ
inant*detpererror,det(a,c,g,k,aa,cc,gg,kk)[1]),
                    multerror(1/determinant,det(c,a,f,d,cc,aa,ff,dd)[0],1/determ
inant*detpererror,det(c,a,f,d,cc,aa,ff,dd)[1])],
                   [multerror(1/determinant,det(d,e,g,h,dd,ee,gg,hh)[0],1/determ
inant*detpererror,det(d,e,g,h,dd,ee,gg,hh)[1]),
                    multerror(1/determinant,det(b,a,h,g,bb,aa,hh,gg)[0],1/determ
inant*detpererror,det(b,a,h,g,bb,aa,hh,gg)[1]),
                    multerror(1/determinant,det(a,b,d,e,aa,bb,dd,ee)[0],1/determ
inant*detpererror,det(a,b,d,e,aa,bb,dd,ee)[1])]])

    return minv, sigmainv

def butter_bandpass(lowcut,highcut,fs,order=5):
    nyq=0.5*fs
    low=lowcut/nyq
    high=highcut/nyq
    b, a = butter(order,[low,high],btype='band')
    return b, a

def butter_bandpass_filter(data,lowcut,highcut,fs,order=5):
    b, a = butter_bandpass(lowcut,highcut,fs,order=order)
    y=lfilter(b, a, data)
    return y

def savitzky_golay(y, window_size, order, deriv=0, rate=1):
    try:
        window_size = np.abs(np.int(window_size))
        order = np.abs(np.int(order))
    except ValueError, msg:
        raise ValueError("window_size and order have to be of type int")
    if window_size % 2 != 1 or window_size < 1:
        raise TypeError("window_size size must be a positive odd number")
    if window_size < order + 2:
        raise TypeError("window_size is too small for the polynomials order")
    order_range = range(order+1)
    half_window = (window_size -1) // 2
    # precompute coefficients
```

```
    b = np.mat([[k**i for i in order_range] for k in range(-half_window, half
_window+1)])
    m = np.linalg.pinv(b).A[deriv] * rate**deriv * factorial(deriv)
    # pad the signal at the extremes with
    # values taken from the signal itself
    firstvals = y[0] - np.abs( y[1:half_window+1][::-1] - y[0] )
    lastvals = y[-1] + np.abs(y[-half_window-1:-1][::-1] - y[-1])
    y = np.concatenate((firstvals, y, lastvals))
    return np.convolve( m[::-1], y, mode='valid')

def calibrate(shot,direction):
    tree=mds.Tree('proto_tree',shot)

    tnode=tree.getNode('\VTIMEVALUES_RCARD1CH0')
    inode=tree.getNode('\VDATAVALUES_RCARD1CH0')
    vxnode=tree.getNode('\VDATAVALUES_RCARD1CH2')
    vynode=tree.getNode('\VDATAVALUES_RCARD1CH6')
    vznode=tree.getNode('\VDATAVALUES_RCARD1CH4')

    mds_time=tnode.getData()
    mds_i=inode.getData()
    mds_vx=vxnode.getData()
    mds_vy=vynode.getData()
    mds_vz=vznode.getData()

    t0=np.array(mds_time)
    rawi=(np.array(mds_i))/0.1
    rawvx=np.array(mds_vx)
    rawvy=np.array(mds_vy)
    rawvz=np.array(mds_vz)


    #sometimes at the end of the shot the voltage jumps, so I cut off the las
t 1/10th of the data
    length=len(t0)
    stop=int(0.9*length)
    time=t0[:stop]
    current=rawi[:stop]
    vx=rawvx[:stop]
    vy=rawvy[:stop]
    vz=rawvz[:stop]


    n=len(time)
```

```
ts=time[1]-time[0]
current_fft = fft.fft(current)
freqs=fft.fftfreq(int(n),ts)
abscurrent_fft=abs(current_fft[1:n/2])
maxvalindex=np.where(abscurrent_fft[:n]==np.amax(abscurrent_fft))[0][0]+1

def my_sin(x, freq, amplitude, phase, offset):
    return np.sin(x * freq + phase) * amplitude + offset

nstd = 20

guess_freq=2*np.pi*freqs[maxvalindex]
guess_phase=0
guess_ioffset=0.5*(np.amax(current)+np.amin(current))
guess_iamp=0.5*(np.amax(current)-np.amin(current))
current_guess=[guess_freq, guess_iamp, guess_phase, guess_ioffset]
ifit=curve_fit(my_sin,time,current,p0=current_guess)
current_fit=my_sin(time,*ifit[0])
shifted_current=current-ifit[0][3]
shifted_current_fit=current_fit-ifit[0][3]
sigmaAcurrent = np.sqrt(np.diag(ifit[1]))[1]


guess_vxoffset=0.5*(np.amax(vx)+np.amin(vx))
guess_vxamp=0.5*(np.amax(vx)-np.amin(vx))
vx_guess=[guess_freq, guess_vxamp, guess_phase, guess_vxoffset]
vxfit=curve_fit(my_sin,time,vx,p0=vx_guess)
vx_fit=my_sin(time,*vxfit[0])
shifted_vx=vx-vxfit[0][3]
shifted_vx_fit=vx_fit-vxfit[0][3]
sigmaAvx = np.sqrt(np.diag(vxfit[1]))[1]
sigmaFvx = np.sqrt(np.diag(vxfit[1]))[0]


guess_vyoffset=0.5*(np.amax(vy)+np.amin(vy))
guess_vyamp=0.5*(np.amax(vy)-np.amin(vy))
vy_guess=[guess_freq, guess_vyamp, guess_phase, guess_vyoffset]
vyfit=curve_fit(my_sin,time,vy,p0=vy_guess)
vy_fit=my_sin(time,*vyfit[0])
shifted_vy=vy-vyfit[0][3]
shifted_vy_fit=vy_fit-vyfit[0][3]
sigmaAvy = np.sqrt(np.diag(vyfit[1]))[1]
sigmaFvy = np.sqrt(np.diag(vyfit[1]))[0]
```

```
    guess_vzoffset=0.5*(np.amax(vz)+np.amin(vz))
    guess_vzamp=0.5*(np.amax(vz)-np.amin(vz))
    vz_guess=[guess_freq, guess_vzamp, guess_phase, guess_vzoffset]
    vzfit=curve_fit(my_sin,time,vz,p0=vz_guess)
    vz_fit=my_sin(time,*vzfit[0])
    shifted_vz=vz-vzfit[0][3]
    shifted_vz_fit=vz_fit-vzfit[0][3]
    sigmaAvz = np.sqrt(np.diag(vzfit[1]))[1]
    sigmaFvz = np.sqrt(np.diag(vzfit[1]))[0]


    def my_cos(x, freq, amplitude, phase):
        return np.sin(x * freq + phase) * amplitude

    vx_int=my_cos(time,vxfit[0][0],-(vxfit[0][1]/vxfit[0][0]),vxfit[0][2])
    vy_int=my_cos(time,vyfit[0][0],-(vyfit[0][1]/vyfit[0][0]),vyfit[0][2])
    vz_int=my_cos(time,vzfit[0][0],-(vzfit[0][1]/vzfit[0][0]),vzfit[0][2])



    mu0=np.pi*4e-7
    nx=4
    ny=4
    nz=5
    ax=.0187
    sigmaax=0.0002

    ay=.02287
    sigmaay=0.00012

    az=.02750
    sigmaaz=0.00015

    if direction == 'x':
        bapplied=(4./5.)**(3./2.)*mu0*nx*shifted_current_fit/ax
        bsigma = (4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaax/ax)**
2+(sigmaAcurrent/ifit[0][1])**2)
    elif direction == 'y':
        bapplied=(4./5.)**(3./2.)*mu0*ny*shifted_current_fit/ay
        bsigma = (4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaay/ay)**
2+(sigmaAcurrent/ifit[0][1])**2)
    elif direction == 'z':
        bapplied=(4./5.)**(3./2.)*mu0*nz*shifted_current_fit/az
```

```
        bsigma =(4./5.)**(3./2.)*mu0*nx*ifit[0][1]/ax*np.sqrt((sigmaaz/az)**2
+(sigmaAcurrent/ifit[0][1])**2)

    bapplied_amplitude = (np.amax(bapplied)-np.amin(bapplied))/2


    grad_vx_int=np.gradient(vx_int)
    grad_vy_int=np.gradient(vy_int)
    grad_vz_int=np.gradient(vz_int)
    grad_current=np.gradient(current_fit)



    if grad_current[0]>0:
        index1=np.where(grad_current<0)[0][0]
    elif grad_current[0]<0:
        index1=np.where(grad_current>0)[0][0]


    if grad_vx_int[index1]>0:
        index2=index1+np.where(grad_vx_int[index1:]<0)[0][0]
    elif grad_vx_int[index1]<0:
        index2=index1+np.where(grad_vx_int[index1:]>0)[0][0]

    if grad_vy_int[index1]>0:
        index3=index1+np.where(grad_vy_int[index1:]<0)[0][0]
    elif grad_vy_int[index1]<0:
        index3=index1+np.where(grad_vy_int[index1:]>0)[0][0]

    if grad_vz_int[index1]>0:
        index4=index1+np.where(grad_vz_int[index1:]<0)[0][0]
    elif grad_vz_int[index1]<0:
        index4=index1+np.where(grad_vz_int[index1:]>0)[0][0]


    xsign = np.sign(vx_int[index2]/bapplied[index1])
    ysign = np.sign(vy_int[index3]/bapplied[index1])
    zsign = np.sign(vz_int[index4]/bapplied[index1])


    xcalfactor = abs(vxfit[0][1]/vxfit[0][0])/bapplied_amplitude
    ycalfactor = abs(vyfit[0][1]/vyfit[0][0])/bapplied_amplitude
    zcalfactor = abs(vzfit[0][1]/vzfit[0][0])/bapplied_amplitude
```

```
    xcalsigma = xcalfactor*np.sqrt((nstd*sigmaAvx/vxfit[0][1])**2+(5*sigmaFvx
/vxfit[0][0])**2+(bsigma/bapplied_amplitude)**2)
    ycalsigma = ycalfactor*np.sqrt((nstd*sigmaAvy/vyfit[0][1])**2+(5*sigmaFvy
/vyfit[0][0])**2+(bsigma/bapplied_amplitude)**2)
    zcalsigma = zcalfactor*np.sqrt((nstd*sigmaAvz/vzfit[0][1])**2+(5*sigmaFvz
/vzfit[0][0])**2+(bsigma/bapplied_amplitude)**2)

    return(xcalfactor,ycalfactor,zcalfactor,xsign,ysign,zsign,xcalsigma,ycals
igma,zcalsigma)


def signmatrix(xshot,yshot,zshot):
    xresponse=calibrate(xshot+1,'x')
    yresponse=calibrate(yshot+1,'y')
    zresponse=calibrate(zshot+1,'z')

    m=[[xresponse[3],yresponse[3],zresponse[3]],[xresponse[4],yresponse[4],zr
esponse[4]],[xresponse[5],yresponse[5],zresponse[5]]]

    return m

def fulltest(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')


    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],zresponse[0]*sig
ns[2][0]],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],zresponse[1]*si
gns[2][1]],
          [xresponse[2]*signs[2][0],yresponse[2]*signs[2][1],zresponse[2]*si
gns[2][2]]]

    sigmainv = [[xresponse[6],yresponse[6],zresponse[6]],
           [xresponse[7],yresponse[7],zresponse[7]],
           [xresponse[8],yresponse[8],zresponse[8]]]

    m,sigmam = matrixinverse(minv,sigmainv)

    return m, sigmam
```

```
def reducedtest(xshot,yshot,zshot,nfreq):
    signs=signmatrix(xshot,yshot,zshot)

    xresponse=calibrate(xshot+nfreq,'x')
    yresponse=calibrate(yshot+nfreq,'y')
    zresponse=calibrate(zshot+nfreq,'z')

    minv=[[xresponse[0]*signs[0][0],yresponse[0]*signs[0][1],1e-20],
          [xresponse[1]*signs[1][0],yresponse[1]*signs[1][1],1e-20],
          [1e-20,1e-20,zresponse[2]*signs[2][2]]]

    sigmainv = [[xresponse[6],yresponse[6],zresponse[6]],
           [xresponse[7],yresponse[7],zresponse[7]],
           [xresponse[8],yresponse[8],zresponse[8]]]

    m,sigmam = matrixinverse(minv,sigmainv)

    return m, sigmam

def currentloop(x,y,z,I,a):
    offset = 0.03
    z=z+offset
    mu0=4*np.pi*10**-7

    rho=np.sqrt(x**2+y**2)
    r=np.sqrt(x**2+y**2+z**2)
    alpha = np.sqrt(a**2+r**2-2*a*rho)
    beta = np.sqrt(a**2+r**2+2*a*rho)
    k = np.sqrt(1-alpha**2/beta**2)
    gamma=x**2-y**2
    C=mu0*I/np.pi



    bx= (C*x*z)/(2*alpha**2*beta*rho**2)*((a**2+r**2)*special.ellipe(k**2)-al
pha**2*special.ellipk(k**2))
    bx=np.nan_to_num(bx)

    by= (C*y*z)/(2*alpha**2*beta*rho**2)*((a**2+r**2)*special.ellipe(k**2)-al
pha**2*special.ellipk(k**2))
    by=np.nan_to_num(by)

    bz= (C/(2*alpha**2*beta))*((a**2-r**2)*special.ellipe(k**2)+alpha**2*spec
ial.ellipk(k**2))
```

```
    np.nan_to_num(bz)



    return (bx,by,bz)

def biascoil(x,y,z,I,a):
    thickness=0.0038
    length = 0.04572
    width = 0.0342
    n=int(np.floor(length/thickness))
    w=int(np.floor(width/thickness))

    bx = 0
    by = 0
    bz = 0

    for i in range(0,n):
        for k in range (0,w):
            bx += currentloop(x,y,z+thickness*i,I,a+thickness*k)[0]
            by += currentloop(x,y,z+thickness*i,I,a+thickness*k)[1]
            bz += currentloop(x,y,z+thickness*i,I,a+thickness*k)[2]

    return (bx,by,bz)

B2_center_12_Vx ='\VDATAVALUES_RCARD0CH0'
B2_center_12_Tx ='\VTIMEVALUES_RCARD0CH0'
B2_center_12_Vy ='\VDATAVALUES_RCARD0CH4'
B2_center_12_Ty ='\VTIMEVALUES_RCARD0CH4'
B2_center_12_Vz ='\VDATAVALUES_RCARD0CH2'
B2_center_12_Tz ='\VTIMEVALUES_RCARD0CH2'

B2_center_13_Vx ='\VDATAVALUES_RCARD0CH6'
B2_center_13_Tx ='\VTIMEVALUES_RCARD0CH6'
B2_center_13_Vy ='\VDATAVALUES_RCARD0CH10'
B2_center_13_Ty ='\VTIMEVALUES_RCARD0CH10'
B2_center_13_Vz ='\VDATAVALUES_RCARD0CH8'
B2_center_13_Tz ='\VTIMEVALUES_RCARD0CH8'

B2_center_14_Vx ='\VDATAVALUES_RCARD0CH12'
B2_center_14_Tx ='\VTIMEVALUES_RCARD0CH12'
B2_center_14_Vy ='\VDATAVALUES_RCARD0CH15'
B2_center_14_Ty ='\VTIMEVALUES_RCARD0CH15'
```

```
B2_center_14_Vz ='\VDATAVALUES_RCARD0CH14'
B2_center_14_Tz ='\VTIMEVALUES_RCARD0CH14'

B2_center_15_Vx ='\VDATAVALUES_RCARD0CH13'
B2_center_15_Tx ='\VTIMEVALUES_RCARD0CH13'
B2_center_15_Vy ='\VDATAVALUES_RCARD0CH9'
B2_center_15_Yy ='\VTIMEVALUES_RCARD0CH9'
B2_center_15_Vz ='\VDATAVALUES_RCARD0CH11'
B2_center_15_Tz ='\VTIMEVALUES_RCARD0CH11'

B1_center_1_Vx ='\VDATAVALUES_RCARD0CH7'
B1_center_1_Tx ='\VTIMEVALUES_RCARD0CH7'
B1_center_1_Vy ='\VDATAVALUES_RCARD0CH3'
B1_center_1_Ty ='\VTIMEVALUES_RCARD0CH3'
B1_center_1_Vz ='\VDATAVALUES_RCARD0CH5'
B1_center_1_Tz ='\VTIMEVALUES_RCARD0CH5'

B1_center_2_Vx ='\VDATAVALUES_RCARD0CH1'
B1_center_2_Tx ='\VTIMEVALUES_RCARD0CH1'
B1_center_2_Vy ='\VDATAVALUES_RCARD0CH18'
B1_center_2_Ty ='\VTIMEVALUES_RCARD0CH18'
B1_center_2_Vz ='\VDATAVALUES_RCARD0CH16'
B1_center_2_Tz ='\VTIMEVALUES_RCARD0CH16'

B1_center_3_Vx ='\VDATAVALUES_RCARD0CH20'
B1_center_3_Tx ='\VTIMEVALUES_RCARD0CH20'
B1_center_3_Vy ='\VDATAVALUES_RCARD0CH24'
B1_center_3_Ty ='\VTIMEVALUES_RCARD0CH24'
B1_center_3_Vz ='\VDATAVALUES_RCARD0CH22'
B1_center_3_Tz ='\VTIMEVALUES_RCARD0CH22'

B1_center_4_Vx ='\VDATAVALUES_RCARD0CH26'
B1_center_4_Tx ='\VTIMEVALUES_RCARD0CH26'
B1_center_4_Vy ='\VDATAVALUES_RCARD0CH30'
B1_center_4_Ty ='\VTIMEVALUES_RCARD0CH30'
B1_center_4_Vz ='\VDATAVALUES_RCARD0CH28'
B1_center_4_Tz ='\VTIMEVALUES_RCARD0CH28'

B1_center_5_Vx ='\VDATAVALUES_RCARD0CH31'
B1_center_5_Tx ='\VTIMEVALUES_RCARD0CH31'
B1_center_5_Vy ='\VDATAVALUES_RCARD0CH27'
B1_center_5_Ty ='\VTIMEVALUES_RCARD0CH27'
B1_center_5_Vz ='\VDATAVALUES_RCARD0CH29'
B1_center_5_Tz ='\VTIMEVALUES_RCARD0CH29'
```

```
B1_center_6_Vx ='\VDATAVALUES_RCARD0CH25'
B1_center_6_Tx ='\VTIMEVALUES_RCARD0CH25'
B1_center_6_Vy ='\VDATAVALUES_RCARD0CH21'
B1_center_6_Ty ='\VTIMEVALUES_RCARD0CH21'
B1_center_6_Vz ='\VDATAVALUES_RCARD0CH23'
B1_center_6_Tz ='\VTIMEVALUES_RCARD0CH23'

B1_center_7_Vx ='\VDATAVALUES_RCARD0CH19'
B1_center_7_Tx ='\VTIMEVALUES_RCARD0CH19'
B1_center_7_Vy ='\VDATAVALUES_RCARD1CH0'
B1_center_7_Ty ='\VTIMEVALUES_RCARD1CH0'
B1_center_7_Vz ='\VDATAVALUES_RCARD0CH17'
B1_center_7_Tz ='\VTIMEVALUES_RCARD0CH17'

B1_center_8_Vx ='\VDATAVALUES_RCARD1CH2'
B1_center_8_Tx ='\VTIMEVALUES_RCARD1CH2'
B1_center_8_Vy ='\VDATAVALUES_RCARD1CH6'
B1_center_8_Ty ='\VTIMEVALUES_RCARD1CH6'
B1_center_8_Vz ='\VDATAVALUES_RCARD1CH4'
B1_center_8_Tz ='\VTIMEVALUES_RCARD1CH4'

B1_center_9_Vx ='\VDATAVALUES_RCARD1CH8'
B1_center_9_Tx ='\VTIMEVALUES_RCARD1CH8'
B1_center_9_Vy ='\VDATAVALUES_RCARD1CH12'
B1_center_9_Ty ='\VTIMEVALUES_RCARD1CH12'
B1_center_9_Vz ='\VDATAVALUES_RCARD1CH10'
B1_center_9_Tz ='\VTIMEVALUES_RCARD1CH10'

B1_center_10_Vx ='\VDATAVALUES_RCARD1CH14'
B1_center_10_Tx ='\VTIMEVALUES_RCARD1CH14'
B1_center_10_Vy ='\VDATAVALUES_RCARD1CH13'
B1_center_10_Ty ='\VTIMEVALUES_RCARD1CH13'
B1_center_10_Vz ='\VDATAVALUES_RCARD1CH15'
B1_center_10_Tz ='\VTIMEVALUES_RCARD1CH15'

B1_center_11_Vx ='\VDATAVALUES_RCARD1CH11'
B1_center_11_Tx ='\VTIMEVALUES_RCARD1CH11'
B1_center_11_Vy ='\VDATAVALUES_RCARD1CH7'
B1_center_11_Ty ='\VTIMEVALUES_RCARD1CH7'
B1_center_11_Vz ='\VDATAVALUES_RCARD1CH9'
B1_center_11_Tz ='\VTIMEVALUES_RCARD1CH9'

B1_center_12_Vx ='\VDATAVALUES_RCARD1CH5'
```

```
B1_center_12_Tx ='\VTIMEVALUES_RCARD1CH5'
B1_center_12_Vy ='\VDATAVALUES_RCARD1CH1'
B1_center_12_Ty ='\VTIMEVALUES_RCARD1CH1'
B1_center_12_Vz ='\VDATAVALUES_RCARD1CH3'
B1_center_12_Tz ='\VTIMEVALUES_RCARD1CH3'

B1_center_13_Vx ='\VDATAVALUES_RCARD1CH16'
B1_center_13_Tx ='\VTIMEVALUES_RCARD1CH16'
B1_center_13_Vy ='\VDATAVALUES_RCARD1CH20'
B1_center_13_Ty ='\VTIMEVALUES_RCARD1CH20'
B1_center_13_Vz ='\VDATAVALUES_RCARD1CH18'
B1_center_13_Tz ='\VTIMEVALUES_RCARD1CH18'

B1_center_14_Vx ='\VDATAVALUES_RCARD1CH22'
B1_center_14_Tx ='\VTIMEVALUES_RCARD1CH22'
B1_center_14_Vy ='\VDATAVALUES_RCARD1CH26'
B1_center_14_Ty ='\VTIMEVALUES_RCARD1CH26'
B1_center_14_Vz ='\VDATAVALUES_RCARD1CH24'
B1_center_14_Tz ='\VTIMEVALUES_RCARD1CH24'

B1_center_15_Vx ='\VDATAVALUES_RCARD1CH28'
B1_center_15_Tx ='\VTIMEVALUES_RCARD1CH28'
B1_center_15_Vy ='\VDATAVALUES_RCARD1CH31'
B1_center_15_Ty ='\VTIMEVALUES_RCARD1CH31'
B1_center_15_Vz ='\VDATAVALUES_RCARD1CH30'
B1_center_15_Tz ='\VTIMEVALUES_RCARD1CH30'

B1_center_16_Vx ='\VDATAVALUES_RCARD1CH29'
B1_center_16_Tx ='\VTIMEVALUES_RCARD1CH29'
B1_center_16_Vy ='\VDATAVALUES_RCARD1CH25'
B1_center_16_Ty ='\VTIMEVALUES_RCARD1CH25'
B1_center_16_Vz ='\VDATAVALUES_RCARD1CH27'
B1_center_16_Tz ='\VTIMEVALUES_RCARD1CH27'

B1_center_17_Vx ='\VDATAVALUES_RCARD1CH23'
B1_center_17_Tx ='\VTIMEVALUES_RCARD1CH23'
B1_center_17_Vy ='\VDATAVALUES_RCARD1CH19'
B1_center_17_Ty ='\VTIMEVALUES_RCARD1CH19'
B1_center_17_Vz ='\VDATAVALUES_RCARD1CH21'
B1_center_17_Tz ='\VTIMEVALUES_RCARD1CH21'

B1_center_18_Vx ='\VDATAVALUES_RCARD1CH17'
B1_center_18_Tx ='\VTIMEVALUES_RCARD1CH17'
B1_center_18_Vy ='\VDATAVALUES_RCARD2CH2'
```

```
B1_center_18_Ty ='\VTIMEVALUES_RCARD2CH2'
B1_center_18_Vz ='\VDATAVALUES_RCARD2CH0'
B1_center_18_Tz ='\VTIMEVALUES_RCARD2CH0'

B1_center_19_Vx ='\VDATAVALUES_RCARD2CH4'
B1_center_19_Tx ='\VTIMEVALUES_RCARD2CH4'
B1_center_19_Vy ='\VDATAVALUES_RCARD2CH8'
B1_center_19_Ty ='\VTIMEVALUES_RCARD2CH8'
B1_center_19_Vz ='\VDATAVALUES_RCARD2CH6'
B1_center_19_Tz ='\VTIMEVALUES_RCARD2CH6'

B1_center_20_Vx ='\VDATAVALUES_RCARD2CH10'
B1_center_20_Tx ='\VTIMEVALUES_RCARD2CH10'
B1_center_20_Vy ='\VDATAVALUES_RCARD2CH14'
B1_center_20_Ty ='\VTIMEVALUES_RCARD2CH14'
B1_center_20_Vz ='\VDATAVALUES_RCARD2CH12'
B1_center_20_Tz ='\VTIMEVALUES_RCARD2CH12'

B1_center_21_Vx ='\VDATAVALUES_RCARD2CH15'
B1_center_21_Tx ='\VTIMEVALUES_RCARD2CH15'
B1_center_21_Vy ='\VDATAVALUES_RCARD2CH11'
B1_center_21_Ty ='\VTIMEVALUES_RCARD2CH11'
B1_center_21_Vz ='\VDATAVALUES_RCARD2CH13'
B1_center_21_Tz ='\VTIMEVALUES_RCARD2CH13'

B1_center_22_Vx ='\VDATAVALUES_RCARD2CH9'
B1_center_22_Tx ='\VTIMEVALUES_RCARD2CH9'
B1_center_22_Vy ='\VDATAVALUES_RCARD2CH5'
B1_center_22_Ty ='\VTIMEVALUES_RCARD2CH5'
B1_center_22_Vz ='\VDATAVALUES_RCARD2CH7'
B1_center_22_Tz ='\VTIMEVALUES_RCARD2CH7'

B1_center_23_Vx ='\VDATAVALUES_RCARD2CH3'
B1_center_23_Tx ='\VTIMEVALUES_RCARD2CH3'
B1_center_23_Vy ='\VDATAVALUES_RCARD2CH16'
B1_center_23_Ty ='\VTIMEVALUES_RCARD2CH16'
B1_center_23_Vz ='\VDATAVALUES_RCARD2CH1'
B1_center_23_Tz ='\VTIMEVALUES_RCARD2CH1'

B1_center_24_Vx ='\VDATAVALUES_RCARD2CH18'
B1_center_24_Tx ='\VTIMEVALUES_RCARD2CH18'
B1_center_24_Vy ='\VDATAVALUES_RCARD2CH22'
B1_center_24_Ty ='\VTIMEVALUES_RCARD2CH22'
B1_center_24_Vz ='\VDATAVALUES_RCARD2CH20'
```

```
B1_center_24_Tz ='\VTIMEVALUES_RCARD2CH20'

B3_center_12_Vx ='\VDATAVALUES_RCARD2CH24'
B3_center_12_Tx ='\VTIMEVALUES_RCARD2CH24'
B3_center_12_Vy ='\VDATAVALUES_RCARD2CH28'
B3_center_12_Ty ='\VTIMEVALUES_RCARD2CH28'
B3_center_12_Vz ='\VDATAVALUES_RCARD2CH26'
B3_center_12_Tz ='\VTIMEVALUES_RCARD2CH26'

B3_center_13_Vx ='\VDATAVALUES_RCARD2CH30'
B3_center_13_Tx ='\VTIMEVALUES_RCARD2CH30'
B3_center_13_Vy ='\VDATAVALUES_RCARD2CH29'
B3_center_13_Ty ='\VTIMEVALUES_RCARD2CH29'
B3_center_13_Vz ='\VDATAVALUES_RCARD2CH31'
B3_center_13_Tz ='\VTIMEVALUES_RCARD2CH31'

B3_center_14_Vx ='\VDATAVALUES_RCARD2CH27'
B3_center_14_Tx ='\VTIMEVALUES_RCARD2CH27'
B3_center_14_Vy ='\VDATAVALUES_RCARD2CH23'
B3_center_14_Ty ='\VTIMEVALUES_RCARD2CH23'
B3_center_14_Vz ='\VDATAVALUES_RCARD2CH25'
B3_center_14_Tz ='\VTIMEVALUES_RCARD2CH25'

B3_center_15_Vx ='\VDATAVALUES_RCARD2CH21'
B3_center_15_Tx ='\VTIMEVALUES_RCARD2CH21'
B3_center_15_Vy ='\VDATAVALUES_RCARD2CH17'
B3_center_15_Ty ='\VTIMEVALUES_RCARD2CH17'
B3_center_15_Vz ='\VDATAVALUES_RCARD2CH19'
B3_center_15_Tz ='\VTIMEVALUES_RCARD2CH19'

def Bfield(shotnum,attenuation,correctionangle):
    angle=correctionangle*np.pi/180

    def rotmatrix(theta):
        return np.array([[np.cos(theta),-1*np.sin(theta),0],[np.sin(theta),np
.cos(theta),0],[0,0,1]])

    tree = mds.Tree('op_tree', shotnum)
    tnode = tree.getNode(B1_center_23_Tx)
    mds_time = tnode.getData()
    time = np.array(mds_time)

    fs=50e6
    lowcut=13000
```

```
    highcut=23e6

    xcalshotlist = [12274,12307,12340,12373,12406,
             12447,12480,12513,12546,12579,12612,12656,12689,12733,12766,1280
4,12837,12870,12903,12936,
             12969,13002,13035,13068,13101]

    ycalshotlist = [12285,12318,12351,12384,12417,
             12458,12491,12524,12557,12590,12634,12667,12700,12744,12777,1281
5,12848,12881,12914,12947,
             12980,13013,13046,13079,13112]

    zcalshotlist = [12296,12329,12362,12395,12428,
             12469,12502,12535,12568,12601,12645,12678,12722,12755,12788,1282
6,12859,12892,12925,12958,
             12991,13024,13057,13090,13123]

    vxcardlist = [B1_center_1_Vx,B1_center_2_Vx,B1_center_3_Vx,B1_center_4_Vx
,B1_center_5_Vx,B1_center_6_Vx,B1_center_7_Vx,
                  B1_center_8_Vx,B1_center_9_Vx,B1_center_10_Vx,B1_center_11_V
x,B1_center_12_Vx,B1_center_13_Vx,B1_center_14_Vx,
                  B1_center_15_Vx,B1_center_16_Vx,B1_center_17_Vx,B1_center_18
_Vx,B1_center_19_Vx,B1_center_20_Vx,B1_center_21_Vx,
                   B1_center_22_Vx,B1_center_23_Vx,B1_center_24_Vx]

    vycardlist = [B1_center_1_Vy,B1_center_2_Vy,B1_center_3_Vy,B1_center_4_Vy
,B1_center_5_Vy,B1_center_6_Vy,B1_center_7_Vy,
                  B1_center_8_Vy,B1_center_9_Vy,B1_center_10_Vy,B1_center_11_V
y,B1_center_12_Vy,B1_center_13_Vy,B1_center_14_Vy,
                  B1_center_15_Vy,B1_center_16_Vy,B1_center_17_Vy,B1_center_18
_Vy,B1_center_19_Vy,B1_center_20_Vy,B1_center_21_Vy,
                   B1_center_22_Vy,B1_center_23_Vy,B1_center_24_Vy]

    vzcardlist = [B1_center_1_Vz,B1_center_2_Vz,B1_center_3_Vz,B1_center_4_Vz
,B1_center_5_Vz,B1_center_6_Vz,B1_center_7_Vz,
                  B1_center_8_Vz,B1_center_9_Vz,B1_center_10_Vz,B1_center_11_V
z,B1_center_12_Vz,B1_center_13_Vz,B1_center_14_Vz,
                  B1_center_15_Vz,B1_center_16_Vz,B1_center_17_Vz,B1_center_18
_Vz,B1_center_19_Vz,B1_center_20_Vz,B1_center_21_Vz,
                   B1_center_22_Vz,B1_center_23_Vz,B1_center_24_Vz]

    zcancel = [1,1,1,1,1,1,1,0,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1]

    xflip = [1,-1,1,1,1,1,-1,1,1,1,1,1,-1,-1,1,1,-1,-1,1,1,1,1,1,1]
```

```
    yflip = [1,-1,1,1,1,-1,1,1,1,1,1,1,-1,1,1,1,1,1,1,1,1,1,1,1]
    zflip = [1,1,1,1,1,-3.6,-1,1,1,1,1,1,-1,1,1,1,1,1,1,1,1,4.5,1,1]

    rawvxlist = [[] for _ in range(24)]
    rawvylist = [[] for _ in range(24)]
    rawvzlist = [[] for _ in range(24)]

    vxlist = [[] for _ in range(24)]
    vylist = [[] for _ in range(24)]
    vzlist = [[] for _ in range(24)]


    integrated_vxlist=[[] for _ in range(24)]
    integrated_vylist=[[] for _ in range(24)]
    integrated_vzlist=[[] for _ in range(24)]


    bx=[[] for _ in range(24)]
    by=[[] for _ in range(24)]
    bz=[[] for _ in range(24)]

    bxsigma=[[] for _ in range(24)]
    bysigma=[[] for _ in range(24)]
    bzsigma=[[] for _ in range(24)]

    bxplus=[[] for _ in range(24)]
    byplus=[[] for _ in range(24)]
    bzplus=[[] for _ in range(24)]

    bxminus=[[] for _ in range(24)]
    byminus=[[] for _ in range(24)]
    bzminus=[[] for _ in range(24)]


    b=[[] for _ in range(24)]
    bplus=[[] for _ in range(24)]
    bminus=[[] for _ in range(24)]

    for i in range(0,24):
        m = fulltest(xcalshotlist[i],ycalshotlist[i],zcalshotlist[i],4)[0]
        msigma= fulltest(xcalshotlist[i],ycalshotlist[i],zcalshotlist[i],4)[1
]
```

```
vxnode=tree.getNode(vxcardlist[i])
vynode=tree.getNode(vycardlist[i])
vznode=tree.getNode(vzcardlist[i])
mds_vx=vxnode.getData()
mds_vy=vynode.getData()
mds_vz=vznode.getData()

rawvx=np.array(mds_vx)
rawvy=np.array(mds_vy)
rawvz=np.array(mds_vz)

tstart=np.where(time*1e6>8)[0][0]
tend=np.where(time*1e6>200)[0][0]
tspan=time[:tend]

offsetx=np.mean(rawvx[:tstart])
offsety=np.mean(rawvy[:tstart])
offsetz=np.mean(rawvz[:tstart])


vx=(rawvx-offsetx)*attenuation
vy=(rawvy-offsety)*attenuation
vz=(rawvz-offsetz)*attenuation*zcancel[i]

vxfilter=butter_bandpass_filter(vx,lowcut,highcut,fs,order=3)
vyfilter=butter_bandpass_filter(vy,lowcut,highcut,fs,order=3)
vzfilter=butter_bandpass_filter(vz,lowcut,highcut,fs,order=3)


vxint=integrate.cumtrapz(vxfilter[:tend],tspan,initial=0)
vyint=integrate.cumtrapz(vyfilter[:tend],tspan,initial=0)
vzint=integrate.cumtrapz(vzfilter[:tend],tspan,initial=0)

rawvxlist[i]=vx
rawvylist[i]=vy
rawvzlist[i]=vz


vxlist[i]=vxfilter
vylist[i]=vyfilter
vzlist[i]=vzfilter


integrated_vxlist[i]=vxint
```

```
        integrated_vylist[i]=vyint
        integrated_vzlist[i]=vzint



        bx[i] = (m[0][0]*vxint*xflip[i] + m[0][1]*vyint*yflip[i] + m[0][2]*vz
int*zflip[i])
        by[i] = (m[1][0]*vxint*xflip[i] + m[1][1]*vyint*yflip[i] + m[1][2]*vz
int*zflip[i])
        bz[i] = (m[2][0]*vxint*xflip[i] + m[2][1]*vyint*yflip[i] + m[2][2]*vz
int*zflip[i])

        bxsigma[i] = np.sqrt((msigma[0][0]*vxint)**2+(msigma[0][1]*vyint)**2+
(msigma[0][2]*vzint)**2)
        bysigma[i] = np.sqrt((msigma[1][0]*vxint)**2+(msigma[1][1]*vyint)**2+
(msigma[1][2]*vzint)**2)
        bzsigma[i] = np.sqrt((msigma[2][0]*vxint)**2+(msigma[2][1]*vyint)**2+
(msigma[2][2]*vzint)**2)

        bxplus[i] = bx[i]+bxsigma[i]
        byplus[i] = by[i]+bysigma[i]
        bzplus[i] = bz[i]+bzsigma[i]

        bxminus[i] = bx[i]-bxsigma[i]
        byminus[i] = by[i]-bysigma[i]
        bzminus[i] = bz[i]-bzsigma[i]

        b[i]=np.dot((rotmatrix(angle)),[bx[i],by[i],bz[i]])
        bplus[i]=np.dot((rotmatrix(angle)),[bxplus[i],byplus[i],bzplus[i]])
        bminus[i]=np.dot((rotmatrix(angle)),[bxminus[i],byminus[i],bzminus[i]
])

    return (b,bplus,bminus,tspan)

chiporder = ['center.1','center.2','center.3','center.4','center.5','center.6
','center.7','center.8','center.9','center.10',
               'center.11','center.12','center.13','center.14','center.15','
center.16','center.17','center.18','center.19','center.20',
               'center.21','center.22','center.23','center.24']

data = Bfield(7150,100,0)
B = data[0]
Bplus = data[1]
```

```
Bminus = data[2]
time = data[3]
```

```
C:\Users\Sander\Anaconda2\lib\site-packages\matplotlib\collections.py:590: Fu
tureWarning: elementwise comparison failed; returning scalar instead, but in
the future will perform elementwise comparison
  if self._edgecolors == str('face'):
```

In [ ]:

```
fig1 = plt.figure(facecolor='white')
for i in range (0,24):
    plt.subplot(6,4,i+1)
    plt.plot(time*1e6,B[i][2],label = 'Bx')
    plt.fill_between(time*1e6,Bminus[i][2],Bplus[i][2],alpha = 0.5,color = 'b
')
    plt.plot(time*1e6,B[i][1],label = 'By')
    plt.fill_between(time*1e6,Bminus[i][1],Bplus[i][1],alpha = 0.5,color = 'g
')
    plt.plot(time*1e6,B[i][0],label = 'Bz')
    plt.fill_between(time*1e6,Bminus[i][0],Bplus[i][0],alpha=0.5,color = 'r')
#    plt.plot(time*1e6,np.sqrt(B[i][2]**2+B[i][1]**2+B[i][0]**2),color = 'bla
ck')



    plt.fill_between(np.array([5,10]),np.array([200,200]),color='0.5',hatch='
O',alpha=0.2)
    plt.fill_between(np.array([5,10]),np.array([-100,-100]),color='0.5',hatch
='O',alpha=0.2)

    plt.fill_between(np.array([10,46]),np.array([200,200]),color='0.5',hatch=
'+',alpha=0.35)
    plt.fill_between(np.array([10,46]),np.array([-100,-100]),color='0.5',hatc
h='+',alpha=0.35)

    plt.fill_between(np.array([46,100]),np.array([200,200]),color='0.5',hatch
='\_/',alpha=0.5)
    plt.fill_between(np.array([46,100]),np.array([-100,-100]),color='0.5',hat
ch='\_/',alpha=0.5)
    plt.axis([-0.025,200,-.5,.5])
#    plt.legend(fontsize = 8)
    plt.tick_params(labelsize=18)
    plt.title(chiporder[i],y=.01)

def lineplots(Bdata,bplus,bminus,timedata,time):
```

```
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .5 #axial location of bdot probe

    chippositions = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.6
,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1])

    chippositionsx = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.
6,
        -3.3,  -2.2,  -1.1 ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   8.8,   9.9,  11. ,  12.1])

    timeindex = np.where(timedata*1e6>time)[0][0]

    t = timedata[timeindex]*1e6

    Barray = np.array(Bdata)
    Bplusarray = np.array(bplus)
    Bminusarray= np.array(bminus)

    biasx=biascoil(chippositions/100,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions/100,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions/100,0,zprobe,I,a)[2]

    Bx =np.delete(Barray[:,2,timeindex]+biasx,(7,8,12,19),0)
    By = Barray[:,1,timeindex]+ biasy
    Bz = Barray[:,0,timeindex]+ biasz

    bxerror = np.delete(Bplusarray[:,2,timeindex]-Bminusarray[:,2,timeindex],
(7,8,12,19),0)/2
    byerror = (Bplusarray[:,1,timeindex]-Bminusarray[:,1,timeindex])/2
    bzerror = (Bplusarray[:,0,timeindex]-Bminusarray[:,0,timeindex])/2


    #    fig = plt.figure(facecolor='white',figsize = (17,13))
    plt.errorbar(chippositionsx,Bx, yerr=bxerror,marker ='.',markersize=20, c
olor = 'b',linewidth =5,elinewidth=3,label = 'Bx')
    plt.errorbar(chippositions,By, yerr=byerror,marker ='.',markersize=20, co
lor = 'g',linewidth =5,elinewidth=3, label = 'By')
    plt.errorbar(chippositions,Bz, yerr=bzerror,marker ='.',markersize=20, co
lor = 'r',linewidth =5,elinewidth=3, label = 'Bz')
```

```
#    plt.plot(chippositionsx,2*np.pi*abs(chippositionsx/100)*Bzp/(1.08*np.sqr
t(Byp**2+Bx**2)))
    plt.axis([-14,14,-0.5,0.5])
#    plt.xlabel('position (cm)',fontsize=20)
#    plt.ylabel('B (Tesla)',fontsize=20)
#    plt.legend(fontsize=20)
    plt.grid(True)
#    plt.title(r't = %d $\mu$ s' %t, fontsize = 45,loc = 'right')
    plt.tick_params(labelsize=30)

fig = plt.figure(facecolor='white')
lineplots(B,Bplus,Bminus,time,40)

def contourplots(Bdata,timedata):
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .75 #axial location of bdot probe


    chippositions = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.6
,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1])

    chippositionsx = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.
6,
        -3.3,  -2.2,  -1.1 ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   8.8,   9.9,  11. ,  12.1])


    X, Y = np.meshgrid(timedata*1e6,chippositions)

    Xp,Yp = np.meshgrid(timedata*1e6,chippositionsx)

    biasx=biascoil(chippositions/100,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions/100,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions/100,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata*1e6,biasz)[1]
```

```
    Barray = np.array(Bdata)

    Bx = np.delete(Barray[:,2,:]+biasxlist,(7,8,12,19),0)#bx
    By = Barray[:,1,:]+biasylist#by
    Bz = Barray[:,0,:]+biaszlist#bz


    fig1 = plt.figure(facecolor='white',figsize = (17,13))
    plt.subplot(3,1,1)
    levels = np.arange(-0.5,.5,0.002)
    cf=plt.contourf(Xp,Yp,Bx,levels=levels)
    plt.contour(Xp,Yp,Bx,colors = 'k')
    plt.axis([0,120,-13.2,12.1])
    plt.tick_params(labelsize=30)
    plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom='on',      # ticks along the bottom edge are off
    top='on',         # ticks along the top edge are off
    labelbottom='off') # labels along the bottom edge are off


    plt.subplot(3,1,2)
    plt.contourf(X,Y,By,levels=levels)
    plt.contour(X,Y,By,colors = 'k')
    plt.axis([0,120,-13.2,12.1])
    plt.tick_params(labelsize=30)
    plt.tick_params(
    axis='x',          # changes apply to the x-axis
    which='both',      # both major and minor ticks are affected
    bottom='on',      # ticks along the bottom edge are off
    top='on',         # ticks along the top edge are off
    labelbottom='off') # labels along the bottom edge are off


    plt.subplot(3,1,3)
    plt.contourf(X,Y,Bz,levels=levels)
    plt.contour(X,Y,Bz,colors='k')
#    cbar = fig1.colorbar(cf)
#    cbar.ax.tick_params(labelsize=20)
    plt.axis([0,120,-13.2,12.1])
    plt.tick_params(labelsize=30)
```

```
def crossection(Bdata,timedata):
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .75 #axial location of bdot probe


    chippositions = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.6
,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1])

    chippositionsx = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.
6,
        -3.3,  -2.2,  -1.1 ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   8.8,   9.9,  11. ,  12.1])


    X, Y = np.meshgrid(timedata*1e6,chippositions)

    Xp,Yp = np.meshgrid(timedata*1e6,chippositionsx)

    biasx=biascoil(chippositions/100,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions/100,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions/100,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata*1e6,biasz)[1]


    Barray = np.array(Bdata)

    Bx = np.delete(Barray[:,2,:]+biasxlist,(7,8,12,19),0)#bx
    By = Barray[:,1,:]+biasylist#by
    Bz = Barray[:,0,:]+biaszlist#bz
    Bthetamag = np.delete(np.sqrt((Barray[:,2,:]+biasxlist)**2+(Barray[:,1,:]
+biasylist)**2),(7,8,12,19),0)
    Byp = np.delete(Barray[:,1,:]+biasylist,(7,8,12,19),0)

    #setup grid for interpolation for streamlines
    y = timedata*1e6
    x =np.linspace(-13.2,12.1,24)
    xi,yi = np.meshgrid(x,y)
```

```
    #interpolate data onto grid
    pxx = Xp.flatten()
    pyx = Yp.flatten()

    px=X.flatten()
    py=Y.flatten()

    pbx =Bx.flatten()
    pby = By.flatten()


    gbx = griddata(zip(pxx,pyx),pbx,(yi,xi),method='linear')
    gby = griddata(zip(px,py),pby,(yi,xi),method='linear')

    fig1=plt.figure(facecolor='white',figsize = (13,17))
    plt.ylabel('time [us]',fontsize=30)
    plt.xlabel('X position [cm]',fontsize=30)
    levels = np.arange(-0.5,.5,0.002)
    cf = plt.contourf(Y,X,Bz,levels=levels)
#    plt.contour(Y,X,Bz,20,colors='k')
    plt.colorbar(cf)
#    plt.contour(Yp,Xp,Bthetamag,colors = 'k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Yp[skip],Xp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwidt
h = 6,scale= .4,scale_units = 'inches')
    lw = 10*np.sqrt(gbx**2+gby**2)/np.sqrt(gbx**2+gby**2).max()
    plt.streamplot(x,y,gbx,-1*gby,density = 4, color = 'r')
    plt.tick_params(labelsize=30)
#    plt.axis([-14,14,20,65])

    fig2=plt.figure(facecolor='white',figsize = (13,17))
    plt.ylabel('time [us]',fontsize=30)
    plt.xlabel('X position [cm]',fontsize=30)
    levels = np.arange(-0.5,.5,0.002)
    cf = plt.contourf(Y,-X,Bz,levels=levels)
#    plt.contour(Y,-X,Bz,colors='k')
    plt.colorbar(cf)
#    plt.contour(Yp,Xp,Bthetamag,colors = 'k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Yp[skip],-Xp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwid
th = 6,scale= .4,scale_units = 'inches')
    lw = 10*np.sqrt(gbx**2+gby**2)/np.sqrt(gbx**2+gby**2).max()
    plt.streamplot(x,-y,gbx,-1*gby,density = 4, color = 'r')
    plt.tick_params(labelsize=30)
```

```
#    plt.axis([-14,14,20,65])
def current(Bdata,timedata,tstart,tend,span):
    mu0=4*np.pi*10**-7
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .75 #axial location of bdot probe
    scaley = span/(tend-tstart)


    chippositions = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.6
,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1])/100

    chippositionsx = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.
6,
        -3.3,  -2.2,  -1.1 ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   8.8,   9.9,  11. ,  12.1])/100

    tstartindex = np.where(timedata*1e6>tstart)[0][0]
    tendindex = np.where(timedata*1e6>tend)[0][0]

    shift = -((timedata[tendindex]*1e6+timedata[tstartindex]*1e6)/2)*scaley


    Y, X = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chipp
ositions)

    Yp,Xp = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chip
positionsx)


    biasx=biascoil(chippositions,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions,0,zprobe,I,a)[2]

    biasxlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx = np.delete(Barray[:,2,tstartindex:tendindex]+biasxlist,(7,8,12,19),0)
#bx
```

```
    By = Barray[:,1,tstartindex:tendindex]+biasylist#by
    Bz = Barray[:,0,tstartindex:tendindex]+biaszlist#bz
    Byp = np.delete(Barray[:,1,tstartindex:tendindex]+biasylist,(7,8,12,19),0
)
    Bzp = np.delete(Barray[:,0,tstartindex:tendindex]+biaszlist,(7,8,12,19),0
)

    gradientbx=np.gradient(Bx,chippositionsx,timedata[tstartindex:tendindex]*
1e6*scaley,edge_order=2)
    gradientby=np.gradient(Byp,chippositionsx,timedata[tstartindex:tendindex]
*1e6*scaley,edge_order=2)

    fig1 = plt.figure()
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,1e6)
    cf = plt.contourf(Xp,Yp,1/mu0*(gradientby[0]+gradientbx[1]),levels=levels
)#add grad bx since db/dt=-db/dy
    plt.colorbar(cf)
    plt.contour(X,Y,Bz,20,colors='k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Xp[skip],Yp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwidt
h = 3,scale= .5,scale_units = 'inches')
    plt.axis([-.08,.02,-span/2,span/2])

    plt.subplot(2,1,2)
    levels2=np.arange(-8,8,0.2)
    cf2 = plt.contourf(Xp,Yp,(gradientby[0]+gradientbx[1])/Bzp*(span/2),level
s=levels2)
    plt.colorbar(cf2)
    plt.contour(X,Y,Bz,20,colors='k')
    plt.axis([-.08,.02,-span/2,span/2])


    fig2 = plt.figure()
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,1e6)
    cf = plt.contourf(Xp,-Yp,1/mu0*(gradientby[0]-gradientbx[1]),levels=level
s)
    plt.colorbar(cf)
    plt.contour(X,-Y,Bz,20,colors='k')
    skip = (slice(None,None,None),slice(None,None,20))
    plt.quiver(Xp[skip],-Yp[skip],Bx[skip],-1*Byp[skip],width = 0.003,headwid
th = 3,scale= .5,scale_units = 'inches')
    plt.axis([-.08,.02,-span/2,span/2])
```

```
    plt.subplot(2,1,2)
    levels2=np.arange(-8,8,0.2)
    cf2 = plt.contourf(Xp,-Yp,(gradientby[0]-gradientbx[1])/Bzp*(span/2),leve
ls=levels2)
    plt.colorbar(cf2)
    plt.contour(X,-Y,Bz,20,colors='k')
    plt.axis([-.08,.02,-span/2,span/2])

current(B,time,54,73,.1)

def currentinterp(Bdata,timedata,tstart,tend,span,radius):
    mu0=4*np.pi*10**-7
    a=0.0381 #bias coil radius
    I = 640 #bias coil current
    zprobe = .75 #axial location of bdot probe
    scaley = span/(tend-tstart)


    chippositions = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.6
,  -5.5,  -4.4,
        -3.3,  -2.2,  -1.1,   0. ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   7.7,   8.8,   9.9,  11. ,  12.1])/100

    chippositionsx = np.array([-13.2, -12.1, -11. ,  -9.9,  -8.8,  -7.7,  -6.
6,
        -3.3,  -2.2,  -1.1 ,   1.1,   2.2,   3.3,   4.4,   5.5,
         6.6,   8.8,   9.9,  11. ,  12.1])/100

    tstartindex = np.where(timedata*1e6>tstart)[0][0]
    tendindex = np.where(timedata*1e6>tend)[0][0]

    shift = -((timedata[tendindex]*1e6+timedata[tstartindex]*1e6)/2)*scaley


    Y, X = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chipp
ositions)

    Yp,Xp = np.meshgrid(timedata[tstartindex:tendindex]*1e6*scaley+shift,chip
positionsx)


    biasx=biascoil(chippositions,0,zprobe,I,a)[0]
    biasy=biascoil(chippositions,0,zprobe,I,a)[1]
    biasz=biascoil(chippositions,0,zprobe,I,a)[2]
```

```
    biasxlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasx)[1]
    biasylist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasy)[1]
    biaszlist = np.meshgrid(timedata[tstartindex:tendindex]*1e6,biasz)[1]

    Barray = np.array(Bdata)

    Bx = np.delete(Barray[:,2,tstartindex:tendindex]+biasxlist,(7,8,12,19),0)
#bx
    By = Barray[:,1,tstartindex:tendindex]+biasylist#by
    Bz = Barray[:,0,tstartindex:tendindex]+biaszlist#bz
    Byp = np.delete(Barray[:,1,tstartindex:tendindex]+biasylist,(7,8,12,19),0
)
    Bzp= np.delete(Barray[:,0,tstartindex:tendindex]+biaszlist,(7,8,12,19),0)

    #setup grid for interpolated values
    npts=100
    #y=timedata[tstartindex:tendindex]*1e6*scaley+shift
    y=np.linspace(timedata[tstartindex]*1e6*scaley+shift,timedata[tendindex]*
1e6*scaley+shift,npts*2)
    x =np.linspace(-13.2,13.2,npts)/100
    xi,yi = np.meshgrid(x,y)

    #interpolate data onto grid
    pxx=Xp.flatten()
    pyx=Yp.flatten()

    px = X.flatten()
    py = Y.flatten()


    pbx =Bx.flatten()
    pby = By.flatten()
    pbz = Bz.flatten()

    gbx = griddata(zip(pxx,pyx),pbx,(xi,yi),method='linear')
    gby= griddata(zip(px,py),pby,(xi,yi),method='linear')
    gbz = griddata(zip(px,py),pbz,(xi,yi),method='linear')

     #note that first term [0] is gradient in rows (now time) and second term
is gradient in columns (now chip position)
    #not sure why the order changed from the previous current command
    #since y points downward, this first gradient in bx is for time descendin
g alon y axis, corresponding to probe transit top to bottom
```

```
    gradientbx=np.gradient(gbx,y,x,edge_order=2)
    gradientby=np.gradient(gby,y,x,edge_order=2)


    fig1 = plt.figure(facecolor='white',figsize=(12,14))
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,5e6)
    cf = plt.contourf(xi,yi,1/mu0*(gradientby[1]+gradientbx[0]),levels=levels
,cmap='bwr')#add grad bx since db/dt=-db/dy
    cb=plt.colorbar(cf)
    plt.contour(xi,yi,gbz,20,colors='k',linewidths=2)
    skip = (slice(None,None,15),slice(None,None,None))
    plt.quiver(xi[skip],yi[skip],gbx[skip],-1*gby[skip],width = 0.003,headwid
th = 3,scale= .5,scale_units = 'inches')
    plt.axis([-.07,.02,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb.ax.tick_params(labelsize=25)

    plt.subplot(2,1,2)
    levels2 = np.arange(-200,20,20)
    cf2 = plt.contourf(xi,yi,(gradientby[1]+gradientbx[0])/gbz*radius,levels=
levels2, cmap='Blues_r')
    cb2=plt.colorbar(cf2)
    plt.contour(xi,yi,gbz,10,colors='k',linewidths=2)
    plt.axis([-.07,.02,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb2.ax.tick_params(labelsize=25)
    plt.tight_layout()


    fig2 = plt.figure(facecolor='white',figsize=(12,14))
    plt.subplot(2,1,1)
    levels=np.arange(-3e7,3e7,5e6)
    cf = plt.contourf(xi,-yi,1/mu0*(gradientby[1]-gradientbx[0]),levels=level
s,cmap='bwr')
    cb=plt.colorbar(cf)
    plt.contour(xi,-yi,gbz,20,colors='k',linewidths=2)
    skip = (slice(None,None,15),slice(None,None,None))
    plt.quiver(xi[skip],-yi[skip],gbx[skip],-1*gby[skip],width = 0.003,headwi
dth = 3,scale= .5,scale_units = 'inches')
```

```
    plt.axis([-.07,.02,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb.ax.tick_params(labelsize=25)

    plt.subplot(2,1,2)
    levels2 = np.arange(-200,20,20)
    cf2 = plt.contourf(xi,-yi,(gradientby[1]-gradientbx[0])/gbz*radius,levels
=levels2,cmap='Blues_r')
    cb2=plt.colorbar(cf2)
    plt.contour(xi,-yi,gbz,10,colors='k',linewidths=2)
    plt.axis([-.07,.02,-span/2,span/2])
    plt.xlabel('X position [cm]',fontsize=25)
    plt.ylabel('Y position [cm]',fontsize=25)
    plt.tick_params(labelsize=25)
    cb2.ax.tick_params(labelsize=25)
    plt.tight_layout()
```

The following Mathematica notebook is a Taylor state equilibria solver, specifically used for the m=1 Taylor double helix.

m=1 Taylor State Solver

Setup Functions

```
Needs["DifferentialEquations`InterpolatingFunctionAnatomy`"]

λ0[n_, a_, h_] = (n^2 * π^2 / h^2 + BesselJZero[1, 1]^2 / a^2)^(1 / 2);
(*first root of m=0 solution as a function of aspectratio*)

k[n_, L_] = n * π / L; (*axial wave number*)

kprime[λ_, k1_] = Sqrt[λ^2 - k1^2];

besseljprime[n_, x_] = D[BesselJ[n, x1], x1] /. x1 → x;
(*function that returns derivative of BesselJ[n,x]*)

coefficientlist[Nmax_] := Table[(Symbol["A" <> ToString@n]), {n, 0, Nmax}] (*produces coefficient list up to Nmax*)

sum[m_, Nmax_, a_, L_, z_, λ_] :=
 N[
  Sum[(Symbol["A" <> ToString@n]) *
     ((I * m / a) * BesselJ[m, kprime[λ, k[n, L]] * a] * Sin[k[n, L] * z] +
       (k[n, L] * kprime[λ, k[n, L]] / λ) * besseljprime[m, kprime[λ, k[n, L]] * a] * Cos[k[n, L] * z]), {n, 1, Nmax}] +
   2 * I * m * (A0) * a^(m - 1) * Exp[I * λ * z]] (*summation term for ψ, m is azimutahl mode number,
 Nmax is the truncated number of terms, a is the radius, L is the length, z is the axial position,
 λ is the eigenvalue*)

det[m_, Nmax_, a_, L_, λ_] := Module[{list, matrix},
  list = {};

  Do[AppendTo[list, sum[m, Nmax, a, L, z, λ]], {z, 0, L, L / (Nmax)}];

  matrix = Normal@CoefficientArrays[list, coefficientlist[Nmax]][[2]];
  Abs[Det[matrix]]
 ] (*m is the azimuthal mode number, Nmax is the truncated number of terms, a is the radius,
 L is the length,
 λ is the eigenvalue. Returns absolute value of the determinant for the complex matrix obtained from
   the truncated radial boundary condition evaluated at Nmax+1 locations along z,
 a minimum value that does not correspond to a spurrious root corresponds to acceptable values of λ*)

gridlines [Nmax_, L_] := N[Table[i, {i, 1, Nmax}] * π / L] (*coordinates of the spurrious roots at λ=nπ/L*)
```

```
solver[m_, Nmax_, a_, L_, λ0_, λf_, spurremove_] :=
 Module[{points, interp, secondderivative, spuriousroots, index, firstrootindex, firstroot, newindex,
   secondrootindex, secondroot},
  points = Sort[Cases[Normal[LogPlot[det[m, Nmax, a, L, λ], {λ, λ0, λf}, Mesh → All, MaxRecursion → 3]],
     Point[pt_] :> pt, ∞]];
  interp = Interpolation[points]; (*interpolating function from the adaptively sampleed points*)
  secondderivative = interp''[points[[ ;; , 1]]];
  (*second derivative of interpolating function evaluated at the λ values in points,
  positive values indicate location of roots*)

  spuriousroots =
   Flatten[
    Table[Position[points[[ ;; , 1]],
      _?(gridlines[Nmax, L][[i]] - spurremove < # < gridlines[Nmax, L][[i]] + spurremove &)],
     {i, 1, Length[gridlines[Nmax, L]]}]];
  (*index locations of values near the spurious roots*)

  index = Complement[Table[i, {i, 1, Length[points]}], spuriousroots];
  (*index values of locations that are not spurious roots*)
  firstrootindex = Position[secondderivative[[index]], _?(# > 0 &)][[1]];
  (*index of first root in second index space*)
  firstroot = points[[index[[firstrootindex]], 1]][[1]];
  (*convert back to the base index and obtain corresponding λ value*)

  newindex = index[[firstrootindex[[1]] + 2 ;;]];
  secondrootindex = Position[secondderivative[[newindex]], _?(# > 0 &)][[1]];
  secondroot = points[[newindex[[secondrootindex]], 1]][[1]];

  {firstroot, secondroot}
 ](*Solves for the first and second roots of λ for the given aspect ratio L/a. Nmax must be > 1/a,
 reduces search volume to λ0<λ1<λ2<λf,
 spurremove identified index locations of values near spurious roots so they are not returned as the
  real roots. For situations where the real root is near the spurious root the value of spurremove
  must be decreased, starting value 0.0005/a works for aspect ratio of 22, can increase for smaller aspect ratios*)
```

```
avals[m_, Nmax_, a_, L_, λ_, fieldstrength_] := Module[{list, matrix, coefficients},
  list = {};

  Do[AppendTo[list, sum[m, Nmax, a, L, z, λ]], {z, 0, L, L / (Nmax)}];
  matrix = Normal@CoefficientArrays[list, coefficientlist[Nmax]][[2]];
  coefficients = LeastSquares[matrix, ConstantArray[fieldstrength, Nmax + 1]];

  coefficients[[1]] = 0;
  (*sets A0 coefficient to zero since it appears to screw up the magnetic field structure*)

  Do[coefficients[[n]] = 0, {n, Round[L / a] - 1, Nmax + 1}];
  (*removes spurriously large coefficeints that seem to show up at or after terms equal to or larger
   than the aspect ratio*)

  coefficients
 ](*Returns the coefficients An in the summation of ψ,
 total energy input sets the strength of the magnetic field*)

ψ[m_, Nmax_, a_, L_, fieldstrength_, spurremove_, helicitysign_] := Module[{λ0, λf, roots, acoefficients, ψ},
  λ0 = 3.11 / a;
  λf = 3.3 / a;

  roots = solver[m, Nmax, a, L, λ0, λf, spurremove];
  acoefficients = avals[m, Nmax, a, L, helicitysign * roots[[1]], fieldstrength];


  ψ =
   (Sum[acoefficients[[n + 1]] * BesselJ[m, kprime[-roots[[1]], k[n, L]] * r] * Sin[k[n, L] * z], {n, 1, Nmax}] +
      acoefficients[[1]] * r^1 * Exp[I * -roots[[1]] * z]) * Exp[I * m * θ];

  {roots, ListPlot[Norm[#] & /@ acoefficients, Joined → True, PlotRange → Full], ψ}
 ](*returns the flux function for the first root for aspect ratio L/a,
 helicity sign makes λ positive or negative*)

convert[x_, y_, z_] =
  Transpose[{{x / Sqrt[x^2 + y^2], y / Sqrt[x^2 + y^2], 0}, {-y / Sqrt[x^2 + y^2], x / Sqrt[x^2 + y^2], 0}, {0, 0, 1}}];
 (*matrix to convert from cylindrical to cartesian coordinates*)
```

Initialize field

```
a = 1;
L = 25;
L / a
energy = 2 * 10^-3;
spurremove = 0.0005 / a;
Nmax = Round[L / a] + 2
m = 1;
helicitysign = -1;
(*returns aspect ratio and Nmax*)
```

```
25
```

```
27
```

```
ψ1 = ψ[m, Nmax, a, L, energy, spurremove, helicitysign]
```

```
ψ1[[1, 1]]
```

```
3.11872
```

```
Bcyl[r_, ϑ_, z_] = {Re[I / r * ψ1[[3]] + 1 / (helicitysign * ψ1[[1, 1]]) * D[ψ1[[3]], r, z]], Re[-D[ψ1[[3]], r]],
   Re[1 / (helicitysign * ψ1[[1, 1]]) * D[ψ1[[3]], {z, 2}] + (helicitysign * ψ1[[1, 1]]) * ψ1[[3]]]};
 (*returns functions for the magnetic field in cylindrical coordinates*)

B[{x_, y_, z_}] = convert[x, y, z].Bcyl[Sqrt[x^2 + y^2], ArcTan[x, y], z];
```

APPENDIX G

CURRENT AND VOLTAGE WAVEFORM CODE

This appendix contains python code used to access and plot the current and voltage waveforms.

```
import MDSplus as mds
import numpy as np
import matplotlib.pyplot as plt
from math import factorial
%matplotlib


def savitzky_golay(y, window_size, order, deriv=0, rate=1):
    r"""Smooth (and opt|ionally differentiate) data with a Savitzky-Golay fil
ter.
    The Savitzky-Golay filter removes high frequency noise from data.
    It has the advantage of preserving the original shape and
    features of the signal better than other types of filtering
    approaches, such as moving averages techniques.
    Parameters
    ----------
    y : array_like, shape (N,)
        the values of the time history of the signal.
    window_size : int
        the length of the window. Must be an odd integer number.
    order : int
        the order of the polynomial used in the filtering.
        Must be less then `window_size` - 1.
    deriv: int
        the order of the derivative to compute (default = 0 means only smooth
ing)
    Returns
    -------
    ys : ndarray, shape (N)
        the smoothed signal (or it's n-th derivative).
    Notes
    -----
    The Savitzky-Golay is a type of low-pass filter, particularly
    suited for smoothing noisy data. The main idea behind this
    approach is to make for each point a least-square fit with a
    polynomial of high order over a odd-sized window centered at
```

```
    the point.
    Examples
    --------
    t = np.linspace(-4, 4, 500)
    y = np.exp( -t**2 ) + np.random.normal(0, 0.05, t.shape)
    ysg = savitzky_golay(y, window_size=31, order=4)
    import matplotlib.pyplot as plt
    plt.plot(t, y, label='Noisy signal')
    plt.plot(t, np.exp(-t**2), 'k', lw=1.5, label='Original signal')
    plt.plot(t, ysg, 'r', label='Filtered signal')
    plt.legend()
    plt.show()
    References
    ----------
    .. [1] A. Savitzky, M. J. E. Golay, Smoothing and Differentiation of
       Data by Simplified Least Squares Procedures. Analytical
       Chemistry, 1964, 36 (8), pp 1627-1639.
    .. [2] Numerical Recipes 3rd Edition: The Art of Scientific Computing
       W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery
       Cambridge University Press ISBN-13: 9780521880688
    """

    try:
        window_size = np.abs(np.int(window_size))
        order = np.abs(np.int(order))
    except ValueError, msg:
        raise ValueError("window_size and order have to be of type int")
    if window_size % 2 != 1 or window_size < 1:
        raise TypeError("window_size size must be a positive odd number")
    if window_size < order + 2:
        raise TypeError("window_size is too small for the polynomials order")
    order_range = range(order+1)
    half_window = (window_size -1) // 2
    # precompute coefficients
    b = np.mat([[k**i for i in order_range] for k in range(-half_window, half
_window+1)])
    m = np.linalg.pinv(b).A[deriv] * rate**deriv * factorial(deriv)
    # pad the signal at the extremes with
    # values taken from the signal itself
    firstvals = y[0] - np.abs( y[1:half_window+1][::-1] - y[0] )
    lastvals = y[-1] + np.abs(y[-half_window-1:-1][::-1] - y[-1])
    y = np.concatenate((firstvals, y, lastvals))
    return np.convolve( m[::-1], y, mode='valid')
```

```
tree = mds.Tree('op_tree', 7150)
tnode = tree.getNode('\NITIMEVALUES_DCARD1CH0')
Inode = tree.getNode('\NIDATAVALUES_DCARD1CH0')
mds_time=tnode.getData()
mds_I = Inode.getData()
time = np.array(mds_time);
total_current=(np.array(mds_I)*531-0.042)*1000
smoothI = savitzky_golay(total_current, 11, 3)



Inodecore = tree.getNode('\NIDATAVALUES_DCARD1CH1')
mds_Icore = Inodecore.getData()
core_current = (np.array(mds_Icore)*236-0.071)*1000
smoothIcore = savitzky_golay(core_current, 11, 3)


tvnode=tree.getNode('\TIMEVALUES_TEK1CH1')
mds_vtime=tvnode.getData()
vtime = np.array(mds_vtime);


vnode = tree.getNode('\VOLTAGEVALUES_TEK1CH1')
mds_Vcore = vnode.getData()
Vcore=np.array(mds_Vcore)
smoothVcore = savitzky_golay(Vcore, 11, 3)

vnodeskin = tree.getNode('\VOLTAGEVALUES_TEK1CH3')
mds_Vskin = vnodeskin.getData()
Vskin = np.array(mds_Vskin)
smoothVskin = savitzky_golay(Vskin, 11, 3)

plt.figure(facecolor='white',figsize=(20,8))
plt.plot(vtime*1e6,Vcore,color='red',linewidth=3)
plt.xlabel('time [us]',fontsize=45)
plt.ylabel('Voltage [kV]',fontsize=45)
plt.axis([-10,100,-8,5])


plt.plot(vtime*1e6,Vskin,color='orange',linewidth=3)


plt.grid(True)
```

```
#plt.fill_between(np.array([5.5,10.5]),np.array([200,200]),color='0.5',hatch=
'O',alpha=0.2)
#plt.fill_between(np.array([5.5,10.5]),np.array([-100,-100]),color='0.5',hatc
h='O',alpha=0.2)

#plt.fill_between(np.array([10.5,45]),np.array([200,200]),color='0.5',hatch='
+',alpha=0.35)
#plt.fill_between(np.array([10.5,45]),np.array([-100,-100]),color='0.5',hatch
='+',alpha=0.35)

#plt.fill_between(np.array([45,90]),np.array([200,200]),color='0.5',hatch='\_
/',alpha=0.5)
#plt.fill_between(np.array([45,90]),np.array([-100,-100]),color='0.5',hatch='
\_/',alpha=0.5)
plt.minorticks_on()
plt.tick_params(labelsize=45)
##plt.legend()##
plt.show()


plt.figure(facecolor='white',figsize=(20,8))
plt.plot(time*1e6,total_current*1e-3,color='black', ls='--',linewidth=3)
plt.xlabel('time [us]',fontsize=45)
plt.ylabel('Current [kA]',fontsize=45)
plt.plot(time*1e6,smoothI*1e-3,color='black',linewidth=5)
plt.axis([-10,120,-100,200])

plt.plot(time*1e6,core_current*1e-3,color='red', ls='-.',linewidth=2)
plt.plot(time*1e6,smoothIcore*1e-3,label = 'Savitzky-Golay filter',color='red
',linewidth=3)

plt.plot(time*1e6,total_current*1e-3-core_current*1e-3,ls=':',color='orange',
linewidth=4)
plt.plot(time*1e6,smoothI*1e-3-smoothIcore*1e-3,color='orange',linewidth=3)

plt.grid(True)
#plt.fill_between(np.array([5,11.5]),np.array([200,200]),color='0.5',hatch='O
',alpha=0.2)
#plt.fill_between(np.array([5,11.5]),np.array([-100,-100]),color='0.5',hatch=
'O',alpha=0.2)

#plt.fill_between(np.array([11.5,35]),np.array([200,200]),color='0.5',hatch='
+',alpha=0.35)
#plt.fill_between(np.array([11.5,35]),np.array([-100,-100]),color='0.5',hatch
='+',alpha=0.35)
```

```
#plt.fill_between(np.array([35,76]),np.array([200,200]),color='0.5',hatch='\_
/',alpha=0.5)
#plt.fill_between(np.array([35,76]),np.array([-100,-100]),color='0.5',hatch='
\_/',alpha=0.5)
plt.minorticks_on()
plt.tick_params(labelsize=45)
##plt.legend()##
plt.show()


plt.figure(facecolor='white',figsize=(20,8))
plt.plot(time*1e6,4*np.pi*10**-7*smoothI/0.004*0.08,color='black',linewidth=5
)
plt.xlabel('time [us]',fontsize=45)
plt.ylabel('Current [kA]',fontsize=45)

plt.axis([-10,120,-2,8])



plt.grid(True)
plt.fill_between(np.array([-10,120]),3.11,20,color='r',alpha=0.2)
#plt.fill_between(np.array([-10,100]),np.array([3.11,3.11]),color='r',alpha=0
.2)


plt.minorticks_on()
plt.tick_params(labelsize=45)
##plt.legend()##
plt.tight_layout()
plt.show()
```

APPENDIX H

FIG CODES

This appendix contains python code used to plot the pressure from FIG output voltage and emission current.

```
import MDSplus as mds
import numpy as np
import matplotlib.pyplot as plt


def plot_pressure(shotnum, BasePressure, label):
    tree = mds.Tree('proto_tree', shotnum)
    tnode = tree.getNode('\NITIMEVALUES_DCARD0CH1')
    Inode = tree.getNode('\NIDATAVALUES_DCARD0CH1')
    Vnode = tree.getNode('\NIDATAVALUES_DCARD0CH2')
    mds_time = tnode.getData()
    mds_I = Inode.getData()
    mds_V = Vnode.getData()
    time = np.array(mds_time)
    Emission_Current = np.array(mds_I)/-514.
    Vsignal = (-1*np.array(mds_V))+0.04
    VoverI = Vsignal/Emission_Current
    firstfew = []
    for i in range (0,20,1):
        listvalue = VoverI[i]
        firstfew.append(listvalue)
    a = 3.597e5
    b = .771
    c =np.average(firstfew) - a*(BasePressure**b)
    Pressure = ((1./a)*(VoverI - c))**(1./b)
    plt.semilogy(time, Pressure, label=label)
    plt.axis([-0.3, 3.0, 1e-8, 3e-2])
    plt.grid(True)
```