Formation of a Sheared Flow Z-Pinch

Raymond Golingo

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

Doctor of Philosophy

University of Washington

2003

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Raymond Golingo

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

_____

Uri Shumlak

Reading Committee:

_____

Uri Shumlak

_____

Thomas R. Jarboe

_____

Brian A. Nelson

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Ardor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from the microform."

Signature_____

Date_____

University of Washington

Abstract

Formation of a Sheared Flow Z-Pinch

by Raymond Golingo

Chair of Supervisory Committee:

Professor Uri Shumlak
Aeronautics & Astronautics

The ZaP experiment has been built to experimentally study the theoretical prediction that sheared flows stabilize plasma instabilities. The experiment uses a coaxial accelerator region coupled with an assembly region to generate Z-pinches with an embedded flow. Previous experiments have seen pinch-like structures which persist while the current is maintained. Theories model the initial plasma of these experiments. This dissertation describes the design of the experimental hardware and diagnostics of the ZaP Flow Z-Pinch experiment at the University of Washington. The plasma velocity is measured with passive spectroscopy. A technique is developed to deconvolve the spectra to obtain axial velocity profiles. Z-pinches which remain quiescent for 1000 instability growth times are measured. During the quiescent period in the magnetic mode amplitude, a sheared flow is present. The shear levels are consistent with the published theory. At the end of the quiescent period, the velocity is uniform and below the value predicted by theory. The results are consistent with the theoretical predictions. A deflagration mode in the accelerator supplies plasma to the Z-pinch during the quiescent period. When the deflagration mode ends, the magnetic mode activity increases. This mode of operation appears to only be limited by the gas injection rate, pumping speed, and driving circuit, not instabilities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CCD: Spectrometer with a CCD on one exit slit and a PMT on the other. This spectrometer is presently being used as a survey spectrometer and monochrometer.

DEFLAGRATION MODE: A process in which neutral gas is ionized and then accelerated in an expansion wave.

FWHM: Full width half max. The width of a Gaussian at half the maximum intensity.

HBW: Half band width. Half of width of a filter at half the maximum intensity.

ICCD: Spectrometer with an image intensified CCD. This spectrometer measures the spectral intensities of 20 parallel chords through the plasma over a small time interval.

IDL: the Interactive Data Language. "IDL is an easy-to-learn, cross-platform application that enables in-depth data analysis through industry-leading visualization."- *Research Systems Inc.*

IDS: Ion Doppler spectrometer. This spectrometer measures the time evolution of the spectral intensities of one chord through the plasma.

IMACON: Imacon fast framing camera. The Imacon fast camera takes photos of the plasma at different times.

IMPACT PARAMETER: shortest distance from the axis to a chord.

LASL EXPERIMENT: Los Alamos Scientific Labs. This abbreviation is used in place of "studies conducted on coaxial guns at the Los Alamos Scientific Labs"

MDSPLUS: a set of tools for performing data acquisition and analysis for pulsed experiments.

MPC: Magnetoplasma compressor. Devices studied in Russia which are similar to coaxial guns.

NORMALIZED M_I: the normalized mode amplitude. The normalized mode is defined as $\min\left(m_i/\max\left(m_0, 0.01\right), 1.0\right)$.

PFN: Pulse forming network. The capacitor bank use to provide a constant current to the experiment.

PMT: Photomultiplier tube.

QSPA: Quasi-steady state plasma accelerator. Experiments performed in Russia which studied the steady state behavior of a coaxial gun.

QUIESCENT PERIOD: The time when all the normalized modes are below an empirical value of 0.2.

SHEARED FLOW: differential axial velocity. When axial velocity, v_z, is a function of the radius.

TREE: MDSplus data structure. The data from an MDSplus is stored in a tree structure.

ZAP: The experiment at the University of Washington studying sheared flow stabilization of Z-pinch instabilities.

ACKNOWLEDGMENTS

The author thanks his wife, Jody, and son, Max, whose patience, support, understanding and encouragement have enabled him the complete his studies. They have endured many nights without me. He also wishes thank his entire family for their help and support throughout his academic career. They have taught me the value of hard work.

He also wishes to express sincere appreciation to everyone who has worked on and are currently working on the ZaP experiment. I have been lucky to have had Uri Shumlak as my adviser and Brian Nelson as the co-PI on the experiment. They always have been always been supportive and are great teachers. I have had the good fortune to work with Ed Crawford and Daniel Den Hartog. They have freely shared their knowledge and experience with me. Special thanks to the ZaP team (I wish I could list everyone). Without their hard work, dedication, enthusiasm, and long hours, these results would not have been possible.

Thank you grandma and grandpa.

Chapter 1

INTRODUCTION

As the world's energy reserves are depleted, new sources of power generation are needed. Fusion energy could provide the world with a clean, unlimited supply of electricity. Before fusion power plants can begin generating electricity, advances in plasma science and the technologies involved with reactors must be made. One of these advances may be using flows to help stabilize plasma confinement schemes. Theoretical calculations show sheared flows may stabilize one of the simplest confinement schemes, the Z-pinch [58]. The ZaP (Z-pinch) experiment at the University of Washington is studying the effect of sheared flows on the stability of a plasma [57, 59]. This work discusses the formation of Z-pinches with a sheared flow.

ZaP is designed similarly to coaxial gun experiments conducted at Los Alamos Scientific Laboratory in the 1960's, the "LASL experiment". The goal of the LASL experiment was to develop a coaxial gun to inject large amounts of plasma into a containment device [26]. It formed pinch like structures which were stable for 100 μs under certain operating conditions. The stability mechanism was not identified, though the authors theorized that the stability could be maintained as long as there is material flow. A sheared flow may explain the pinch stability.

While ZaP is similar to the LASL experiment there are many differences. The LASL gun used the vacuum vessel for the outer electrode and fired into a large vacuum chamber. In ZaP, the outer cylindrical electrode (distinct from the vacuum vessel) extends 50 cm beyond the inner electrode and the Z-pinch attaches to an electrode end plate. ZaP uses modern vacuum, manufacturing, computing, and diagnostic technology to improve operation over that of the LASL experiment and to better understand the stability of the Z-pinch.

1.1 Fusion Energy

One of the main purposes of studying magnetic confinement of plasma is energy generation. The world has a finite reserve of fossil fuels, and there is a predicted energy shortage in the middle of the 21st century. Fusion energy would provide the world with a new clean energy source. There is enough fuel (D,T) on earth to meet our energy needs for earth's lifetime. The ash from fusion energy generation is safe, He. The high energy neutrons produced can be used to create T or can be absorbed by materials that are not activated. Fusion must keep its promise to be the cleanest energy source to be a viable energy alternative.

Fusion Reactions

Fusion is the process of combining of two low mass nuclei into a higher mass nucleus whose mass is less than the sum of the original nuclei. One of the by-products of the reaction is energy, given by Einstein's famous equation, $E = mc^2$. The reactions which are normally seen in fusion reactor studies are

$$\begin{aligned} D + D &\overset{50\%}{\rightarrow} T(1.01MeV) + p(3.02MeV) \\ &\overset{50\%}{\rightarrow} He^3(0.82MeV) + n(2.45MeV) \\ D + T &\rightarrow He^4(3.5MeV) + n(14.1MeV) \\ D + He^3 &\rightarrow He^4(3.6MeV) + p(14.7MeV) \\ T + T &\rightarrow He^4 + 2n + 11.3MeV \\ He^3 + T &\overset{51\%}{\rightarrow} He^4 + p + n + 12.1MeV \\ &\overset{43\%}{\rightarrow} He^4(4.8MeV) + D(9.5MeV) \\ &\overset{6\%}{\rightarrow} He^5(2.4MeV) + p(11.9MeV) \end{aligned} \tag{1.1}$$

The most attractive is the D+T reaction, as it has the largest reaction cross section. One of the by-products of this reaction is a 14.1 MeV neutron. The neutron can be used for T production by using Li in the first wall. Since the D+D reaction also produces neutrons, only H_2 is used in ZaP.

Lawson Criteria

A reactor is only attractive if there is net energy production. Before these reactions can occur, particles must have enough energy to overcome the Coulomb barrier between them. For this to occur, a plasma with number density (n), particle confinement time (τ_E), and ion temperature (T_i) equal to the electron temperature (T_e) must satisfy the Lawson criteria [34]

$$n\tau_E > \frac{12}{Q_\alpha}\left(\frac{T_i}{\overline{\sigma v}}\right) \mathrm{m}^{-3}\mathrm{s} \tag{1.2}$$

where Q_α is the α-particle energy and $\overline{\sigma v}$ is the fusion cross section. The D+T reaction cross section has a maximum at $T_i \approx 15$ keV. Reactors which operate at this temperature would then have to meet

$$\begin{aligned} \beta\tau_E &> \frac{2.3}{B^2} \quad (\mathrm{s\ T}^2) \quad \text{or} \\ 2\mu_0 n(T_e + T_i)\tau_E &> 2.3 \quad (\mathrm{s\ T}^2) \quad \text{if } \beta = 1 \end{aligned} \tag{1.3}$$

where $\beta \equiv 2\mu_0 n(T_e + T_i)/B^2$ and B is the magnitude of the magnetic field. Ideal reactors would have the largest β possible, to minimize the applied magnetic field. Stability considerations place limits on β in most devices studied today. This requires τ_E to be large. Relaxing the constraints on β would benefit future projects and reactor designs. One method of showing the stability constraints can be relaxed is to stabilize a high β configuration, the Z-pinch.

1.2 Z-pinches

Some of the first magnetically confined plasmas were Z-pinches [23]. The force balance of this simple confinement scheme, is not affected by axial velocities. Static Z-pinches are unstable to two MHD modes. Different methods have been used to stabilize these modes, which limit the temperature and pressure. Recent theoretical predictions have shown sheared flows may stabilize these modes. The simple geometry and well-documented instabilities make the Z-pinch a good configuration to study the effect of sheared flows on stability.

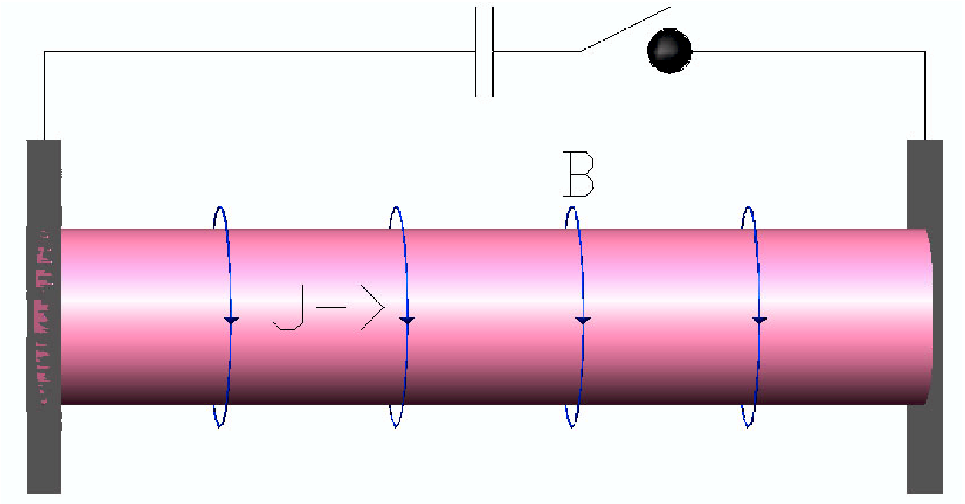

Figure 1.1: Z-pinches are a simple magnetic confinement scheme. A column of plasma (pink) is confined between two electrodes (grey). A capacitor bank and switch (black) are used to drive an axial current, J, through the column of plasma. The self-generated azimuthal magnetic field, B, confines the plasma.

1.2.1 Z-pinch Equilibrium

A Z-pinch is one of the simplest magnetic confinement configurations. It consists of an axial current where the self generated azimuthal magnetic field provides confinement, shown in Fig. 1.1. Since it has a simple geometry, a flow through Z-pinch is ideal for the study of sheared flow stabilization.

The Z-pinch equilibrium can be described with the MHD force balance equation.

$$\nabla P + \rho \left(\mathbf{v} \cdot \nabla \right) \mathbf{v} = \mathbf{J} \times \mathbf{B} \tag{1.4}$$

where P is the pressure, ρ is the density, $\mathbf{v}$ is the velocity, $\mathbf{J}$ is the current density and $\mathbf{B}$ is the magnetic field. For a sheared flow Z-pinch the following assumptions can be made

$$\begin{aligned} \mathbf{v} &= [0, 0, v_z(r)] \\ \mathbf{B} &= [0, B_\theta(r), 0] \\ \mathbf{J} &= [0, 0, J_z(r)] \\ P &= P(r) \ . \end{aligned} \tag{1.5}$$

The axial velocity has no effect on the equilibrium, since $(\mathbf{v}\cdot\nabla)\,\mathbf{v} = \mathbf{0}$. The pressure is described by static Z-pinch equilibrium equation

$$\frac{d}{dr}\left(P + \frac{B_\theta^2}{2\mu_0}\right) + \frac{B_\theta^2}{\mu_0 r} = 0\ . \tag{1.6}$$

A solution of which, given by Bennett [19], is

$$\begin{aligned} B_\theta &= \frac{\mu_0 I_0}{2\pi}\frac{r}{r^2 + r_0^2} r B_\theta \\ J_z &= \frac{I_0}{\pi}\frac{r_0^2}{(r^2 + r_0^2)^2} \\ P &= \frac{\mu_0 I_0^2}{8\pi^2}\frac{r_0^2}{(r^2 + r_0^2)^2} \end{aligned} \tag{1.7}$$

where I_0 is the current through the Z-pinch and r_0 is the characteristic radius. The equilibrium in ZaP is more complicated than this simple relationship because the density must vary along the z axis. At the end of the inner electrode the plasma is accelerated inward adding 2-D effects to the equilibrium. Near the end wall electrode, the velocity stagnates as mass accumulates over time. These effects are minimized at the midplane of the Z-pinch where most of the measurements are made. The time evolution of a flow-through pinch has been simulated with MACH2, a 2-D resistive MHD code [47]. Future work with MACH2 and new codes which model ionization in the accelerator will give a better understanding of the equilibrium and sustainment of the Z-pinch.

1.2.2 Traditional Stabilization Methods

Although the Z-pinch equilibrium is an attractive configuration for fusion, Z-pinches are unstable to the $m = 0$ and $m = 1$ modes, shown in Fig. 1.2 [22]. Many techniques; close fitting walls, axial magnetic fields and pressure profile control; have been used to stabilize Z-pinches. Each has had success in stabilizing the Z-pinch, but plasmas of thermonuclear interest are difficult if not impossible to obtain with these techniques.

A close-fitting wall has been shown to stabilize the Z-pinch [58]. Arc jets are an example of a device which uses close-fitting walls as one of its stabilizing mechanisms. The main problem with the use of close-fitting walls is that the temperature of the plasma is limited by the maximum temperature of the walls. The small area of the wall must take the

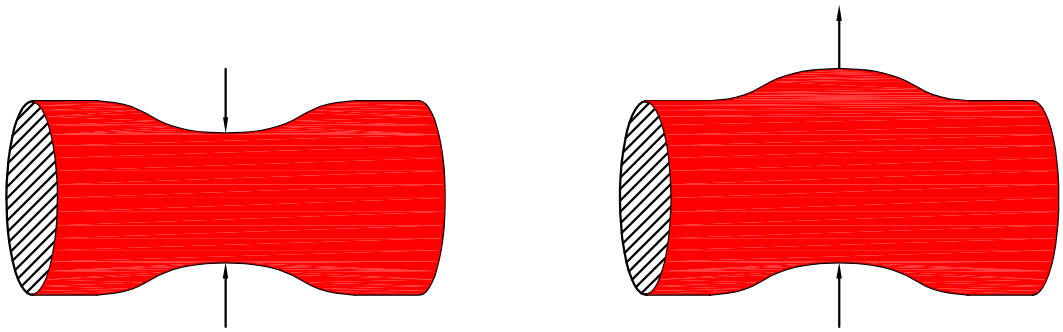

Figure 1.2: Z-pinches are unstable to the sausage or $m = 0$ mode (left side) and the kink or $m = 1$ mode (right side). The $m = 0$ mode is a symmetric necking of the plasma. The $m = 1$ mode is an asymmetric bending of the plasma. The arrows show the direction of the force on the plasma. The perturbations grow until the current is disrupted.

full heat load from the Z-pinch, limiting the maximum edge temperature. This in turn limits the temperature in the core of the plasma. In a thermonuclear plasma, the required temperature is about 10-20 keV. There are no materials which could withstand the heat load from a plasma of this temperature.

Applying an axial magnetic field to the Z-pinch is another stabilization method which has been used. An $m = 0$ mode would compress the axial magnetic field. An $m = 1$ mode would try to bend the axial magnetic field. The tension in the field lines acts as a restoring force on the plasma. The axial plasma current is limited by the stabilizing axial field according to the Kruskal-Shafranov limit, setting an upper bound on the plasma pressure [33, 54]. The confinement time of the plasma is also degraded with the application of an axial magnetic field. Plasma is not confined along magnetic field lines. The axial magnetic field provides a path along which plasma can leave the core. Wrapping the magnetic field into a torus solves the confinement problem, but there still is a limit on the current.

The m=0 mode can be stabilized with a gradual pressure profile according to Kadomstev [19]. The pressure must satisfy the relationship

$$-\frac{d\ln P}{d\ln r} \leq \frac{4\gamma}{2+\gamma\beta} \ . \tag{1.8}$$

where γ is the ratio of specific heats and $\beta = 2\mu_0 P/B^2$. While these plasmas are marginally

stable to the $m = 0$ mode, the stability of the $m = 1$ mode is unaffected. This result is important because the $m = 0$ mode cannot be stabilized with a vacuum boundary [52].

1.2.3 Sheared Flow Stabilization

A better method of stabilization would not limit the pressure nor temperature in the core of the Z-pinch. Sheared flows can stabilize the sausage and kink mode without limiting the pressure or temperature. The coherence of the mode is disrupted with a sheared flow. In a static Z-pinch, the equilibrium is provided by the magnetic pressure $B^2/2\mu_0$ on the outer surface. A small decrease in the radius increases the magnetic pressure at that location accelerating the edge inward. In a static pinch the edge continues to move inward disrupting the current. If a sheared flow is present, a displacement of the edge of the plasma will move into a region with a lower velocity. The axial translation of the displaced edge is slowed as it moves inward. Faster moving plasma from behind the displacement fills in the volume above the perturbation. The coherence of the displacement is disrupted, stabilizing the mode. A uniform velocity shear of

$$\frac{dV_z}{dr} \geq 0.1kV_A, \tag{1.9}$$

where V_z is the axial velocity, k is the wave number and V_A is the Alfvén velocity, would stabilize the kink mode [58].

1.3 Outline of Work

The results from previous experiments are described in Chapter 2. The theories which have been developed from other devices can be used to understand the plasma in the ZaP experiment. The theoretical predictions which have lead to the ZaP experiment are also discussed in this Chapter. The formation process used to create a sheared flow Z-pinch is described in Chapter 3. Snowplow models, which describe the initial formation process, are developed in this Chapter. The design of the major systems on the experiment are described in Chapter 4. These systems include the vacuum, gas feed, energy storage and safety systems. The diagnostic and analysis techniques are described in Chapter 5. A

new technique to calculate the plasma properties from passive spectroscopy is described. Chapter 5 shows the results from four studies done on the experiment. The discussion of the results is in Chapter 6. The plasma characteristics are compared to the theories present in the previous Chapters. A heuristic explanation of the formation and sustainment processes is also presented. Chapter 7 discusses the conclusions from this work. Possible future work on the ZaP experiment is presented in Chapter 8. Practical applications of a sheared flow Z-pinch are discussed in the Chapter.

Appendix A discusses a technique to calculate the magnetic at the surface of a flux conserver. The magnetic fields are found without calculating the image currents. Gas puff studies used to find the optimum gas puff timings are shown in Appendix B. The codes used to analyze the data are found in Appendix C.

Chapter 2

PREVIOUS WORK

The study of sheared flow stabilization requires a unique machine to generate the flows required to stabilize the MHD modes. ZaP uses a coaxial gun to initiate the Z-pinch plasma. Different phases of the evolution of the plasma have been studied in other experiments. Coaxial guns and plasma focus devices have developed the theories needed to explain the acceleration of the current sheet and compression of the plasma near the inner electrode. A period of MHD stability has been reported in these devices.

Z-pinch plasmas were some of the first plasmas to be studied. As shown in the previous section, Z-pinches have a simple magnetic topology. Present research uses Z-pinches which are pulsed, short-lived plasmas. The Rayleigh-Taylor instabilities have been mitigated by using wire arrays. These Z-pinches are used as a sources of hot dense plasma, intensive x-rays, and ultra high magnetic fields [44]. Since Z-pinches are a simple configuration, and the instabilities are well understood, they make an ideal configuration to study sheared flow stabilization.

Early work postulated that flows may stabilize a plasma [26]. The discovery of transport barriers in tokamak plasmas renewed interest in plasma flows. The H-mode transport barrier is the most studied. There is substantial evidence that the velocity shear is partially responsible for the internal transport barrier [62]. The experimental evidence of flow stabilization has lead to computer simulations which include the dynamics of the plasma [68]. The evidence that adding energy to a plasma can positively influence the behavior lead to an increased interest in added flows to the MHD equations. New theoretical predictions showed that the MHD modes which disrupt a Z-pinch can be stabilized with a sheared flow. This prediction is presently being investigated by the ZaP experiment.

2.1 Coaxial guns

Coaxial guns were designed to accelerate plasma to high velocities. They inject gas into the annulus between the electrodes. A high voltage is applied to the electrodes with large capacitor banks. The gas is ionized, forming a current sheet, which is accelerated out of the gun. This work was done mainly by Marshall *et al.* at Los Alamos and Cheng at Santa Clara [14, 37]. Coaxial gun research decreased in the USA in the early 1970's. Morozov *et al.* built and studied a magnetoplasma compressor in 1967 [42]. A steady state coaxial plasma accelerator was studied in the late 1980's to the early 1990's at the Kurchatov Institute Russian Research Center.

2.1.1 Los Alamos Studies

In 1959, Marshall discovered that he could pass a 200 kA current through the gun without detaching material from the electrodes [37]. Two different plasmas were defined in the gun, the "fast plasma" and the "slow plasma". The fast plasma, which was entrained in the current sheet, could be used to fuel other machines. A snowplow model was used to describe this plasma. The slow plasma was pushed against the outer electrode during the acceleration of the current sheet. A continuous flow-through pinch was a result of the slow plasma leaving the annulus of the gun. The flow-pinch persisted in front of the gun for about 100 μs, the energy storage time of the bank [26]. The stability mechanism for this pinch was not found. This result, the flow-through pinch, has laid the groundwork for the ZaP experiment.

Gun Dimensions

The Newton Marshall experiments were done in a 1 m diameter, 2 m long vacuum chamber The inner electrode was a 1.02 m long 6.4 cm outer diameter copper tube. The outer electrode was a 1.02 m long 17.1 cm inner diameter copper tube. The outer electrode was removed for the snowplow studies done by T.D. Butler [9, 10]. An explosive valve was used to inject gas into the annulus between the electrodes. The valve had a 3.3 cm^3 plenum filled to 500 psig. The 2800 μF capacitor bank was fired 400 μs after the valve was opened. It

took the current sheet 15 μs to reach the end of the gun. The plasma was then free to empty into the vacuum chamber. A stable pinch lasted for 100 μs after the current sheet imploded on axis. The pinch had a number density of 10^{17} cm^{-3} and a temperature of 100 eV. The pinch was stable under the appropriate operating conditions.

Models

Marshall *et al.* discovered a fast plasma is created and travels with the current sheet. This plasma is what has been studied in plasma focus experiments and for injection into other devices. The plasma was modeled as a snowplow by T.D. Butler *et al.* [11, 9, 10]. This model uses a massless magnetic piston to accelerate the plasma. The current is in a thin layer between the gas and the plasma. As the current sheet accelerates towards the exit of the gun, a shock forms in front of it. Gas entering the shock is ionized and pushed towards the outer electrode in a thin layer between the shock and the current sheet. This model is explained in Sec. 3.2.2.

Marshall *et al.* further discovered a slow plasma is comprised of most of the gas admitted from the valve. As plasma exits the gun it travels radially inward. A one dimensional model was used to describe the plasma at this location of the gun. Flow along the axis was modeled as a sink for the mass and magnetic flux. The treatment of the slow plasma was different than Morozov's treatment of a similar plasma. The possibility of shocks is not considered in Morozov's model. A shock was seen in the LASL experiment's PIC simulations at the end of inner electrode. Evidence of a shock at the end of the inner electrode was seen during the experiment. While PIC simulations showed structures seen in the current sheet and average properties of the pinch, they were not able to look at the structures inside of the pinch because of limits on their cell size.

2.1.2 University of Santa Clara Studies

D.Y. Cheng was interested in accelerating plasma using coaxial guns [14]. He described the plasma with another process which may be occuring in a coaxial gun. The plasma was modeled similar to the combustion of a gas. The snowplow was described as a detonation

process, where the maximum speed of the plasma was always slower than the speed of the compression wave. He proposed using a deflagration process, where the deflagration wave front travels in the opposite direction as the plasma. The plasma is accelerated in an expansion wave. The final speed of the plasma is not limited by the speed of the compression front.

A deflagration wave was created by charging the electrodes of the gun. Gas was then puffed into the end of the gun. When the pressure was above the Paschen minimum, the gas ionized forming a plasma. The plasma was accelerated to the muzzle of the gun in the expansion wave. The process was sustained by continuously injecting gas into the breakdown region. The directed energy of his plasma was 10 keV to 30 keV. His results could not be confirmed by the LASL experiment.

2.1.3 Kurchatov Institute of Atomic Energy

The basic theory for plasma accelerators were formulated by A.I. Morozov *et al.* in 1959 [42]. Compressional flows were verified experimentally in 1967 in a magnetoplasma compressor (MPC). Instabilities were found in the ionization zone and at the anode of these devices. A two step quasistationary plasma accelerator (QSPA) was studied in the 1990's. Theoretical models were developed which explain the observed phenomena.

An MPC uses a coaxial gun with shaped electrodes [7]. The outer electrode is twice as long as the inner electrode. An MPC operating with hydrogen in 1968 had a steady compressional flow for 100 μs, comparable to the bank discharge time. The plasma had $n_e \approx 6 \times 10^{16}$ cm^{-3} and $T_i \approx T_e \approx 10$ eV with a current of 300 kA.

The two step QSPA is a complicated machine, as seen in Fig. 2.1. In a two step QSPA, plasma is injected into the annulus with an array of modified MPC's. The accelerator is a coaxial gun with shaped rod electrodes. "Current collecting pins" are used in the cathode inner electrode. The anode outer electrode is surrounded by a dielectric screen to prevent the spreading of the injected gas. A 5600 μF capacitor bank, charged to 10 kV, is used to supply the current. The gun is fired into a large vacuum chamber.

The QSPA's were designed to avoid problems seen in earlier devices. A rotational (spin)

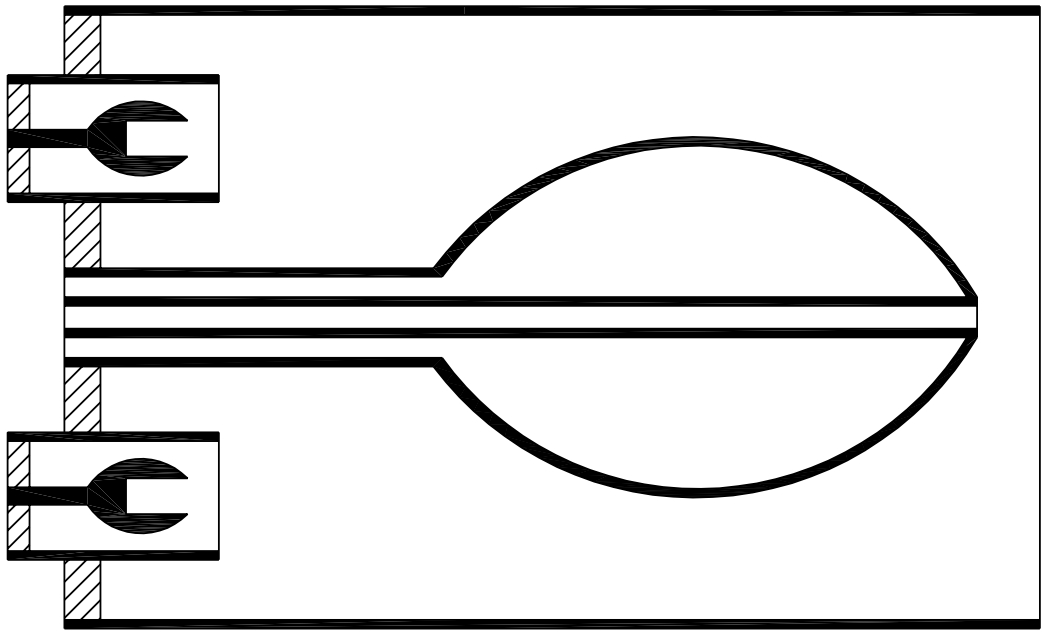

Figure 2.1: Diagram of the P-50 QSPA showing the geometry of the electrodes and the input ionization chambers. The input chambers, shown on the left, are an array of MPCs with solid electrodes. These inject plasma into a larger MCP which has rod electrodes. The inner electrodes are shaped such that plasma exiting the MPCs is supersonic.

instability and a longitudinally oscillatory instability were observed in the ionization zone in the one step QSPA. These instabilities do not occur in a two step QSPA because the gas in preionized in the MPC's. Another effect called "anode current creep" was also observed. This leads to a strong attachment at the anode which is a feature of essentially all QSPA's with solid electrodes. It is caused by a depletion of ions near the anode. Using rod electrodes where plasma passes through the electrodes lessens the "current crisis". Erosion along the electrode surfaces is caused by the ion current in the accelerator. Transformers are used as ion emitters at the anode and as collectors at the cathode. This protects the electrodes and provides a source of ions along the anode, solving the "current crisis" problem.

The velocity and compression of a plasma in a coaxial gun can be understood with the MHD Bernoulli equation. The plasma is partitioned into narrow coaxial tubes bounded by

streamlines $\xi = const$. The properties of the flow in each tube can be described with

$$W(\rho) + \frac{v^2}{2} + \rho r^2 B^2 = U = const \tag{2.1}$$

where $W(\rho) = \int (dP/\rho)$ [43]. In a constant radius channel the maximum velocity is given by $\sqrt{2\left(c_{T0}^2/(\gamma-1) + c_{A0}^2\right)}$, where c_{T0} and c_{A0} are the sound and Alfvén speed at the channel entrance. A varying channel cross-sectional area is needed to accelerate the flow above the signal speed, $c_s = \sqrt{c_T^2 + c_A^2}$. The critical section of the flow, where a subsonic flow transitions to a supersonic flow, occurs near the minimum area of the channel. In devices where the compression of the plasma is important, velocities above the signal speed must be avoided to stop shocks from being generated. The maximum density ratio of an MPC is given by

$$\nu_{max} \equiv \frac{\rho_{max}}{\rho_0} = \left[(\gamma-1)\frac{c_{A0}^2}{c_{T0}^2}\left(1 + \frac{W_0}{c_{A0}^2}\right)\right]^{1/(\gamma-1)} \tag{2.2}$$

These equations were used to design the MPC's and QSPA's.

Computer simulations of MPC's and QSPA's agree with the analysis from the MHD Bernoulli equation. The plasma was modeled with magnetogasdynamics taking into account the Hall effect [6]. Computer simulations using the two-fluid model including transport properties modeled with Braginskii's work have also been done [7]. The simulations have shown that current vortices can form in the QSPA's. Current vortices, measured on the QSPA, are seen shortly after the plasma exits the gun and persist until the end of the shot [65, 66]. Ionization processes have been included in the simulations by using Saha's formula. The longitudinally oscillatory instability was seen during these simulations.

2.2 Plasma Focus Devices

Plasma foci were first studied independently in the USA and the Soviet Union in the 1960's [70]. Mather discovered a dense high temperature plasma in front of the coaxial gun when he operated under certain conditions [39]. When these devices operated with deuterium they produced neutrons. The scaling for the neutron production is ($Y \sim E^2$), where Y is the neutron yield and E is the energy in the capacitor banks.

Plasma foci are normally small devices. The radius of the outer electrode is normally less than 10 cm. The electrodes are less then 30 cm long. The capacitor banks are less than 100 μF. By keeping the external inductance low, less than 100 nH, the quarter period of the current is 0.8 to 4 μs. Large peak currents, from 250 kA to 4 MA, are used to compress the plasma in these devices. The focus lifetime is short, less than 100 ns. The main diagnostics used on plasma focuses are voltage monitors, Rogowski coils, neutron and x-ray detectors. Some experiments have also used magnetic probes, differential interferometry, and Schlieren photography to understand the plasma. The other plasma quantities of interest are then found using models of the plasma.

2.2.1 Formation

A plasma focus is formed with a coaxial gun. The inner electrode is the anode. The outer electrode, the cathode, is about the same length as the inner electrode. The vacuum vessel is filled with 0.5-10.0 mbar of deuterium or a gas mixture. A high voltage from a capacitor bank is applied to the electrodes. The gas breaks down across the insulator forming a current sheet which is accelerated by the $\mathbf{J} \times \mathbf{B}$ force to the muzzle of the gun. The current sheet should reach the exit of the gun when the current peaks for maximum compression of the pinch. The time at which the current sheet reaches the muzzle can be found with

$$t_a = \left(\frac{4\pi^2 z_0^2 (b^2 - a^2) \rho_0}{I_0^2 \mu \ln(b/a)} \right)^{1/2} \tag{2.3}$$

where z_0 is the length of the anode, I_0 is the maximum current, a and b are cathode and anode radii, and ρ_0 is density.[35] There is current shedding as the current sheet travels between the electrodes. Mathuthu reported 68.5% of the total current was in the focus of his device [40]. The process of shedding current, as the current sheet travels along the inner electrode, is a transition from the snowplow to deflagration description of the current in the gun [69]. After the current passes the anode it collapses on axis with a zippering action forming the plasma focus. This zippering action compresses the plasma to a high density and temperature. During this time a plasma jet has been seen in front of the current sheet [17]. The focus then goes unstable from the growth of the $m = 0$ mode in the pinch, or from radiative collapse. Neutrons are produced during this time.

2.2.2 Neutron Production

Plasma focus experiments are mainly looking at ways to produce neutrons and x-rays. The neutrons are produced from two different mechanisms in the focus [13]. There are thermal neutrons produced in the plasma and beam-target neutrons produced when jets from the plasma impact a surface or medium. Various inner electrode shapes have been tried. Zakaullah *et al.* found that a tapered electrode produced the most x-ray radiation [70]. Lu has identified an unstable mode during the formation of a plasma focus using a solid inner electrode [36]. This mode occured on 44% of his shots. The mode occurs when the angle between the current sheet and the vertical electrode face is small. The neutron production during the unstable shots is about half the production of neutrons on a stable shot. This mode is not present when a hollow electrode is used.

Most of the neutrons are produced after the plasma focus has reached its maximum compression and has gone unstable. A plasma focus in Germany, SPEED2, has found an operating region where they can produce neutrons before maximum compression in a stable pinch [31]. SPEED2 injects gas along the z axis. When argon is injected the plasma focus forms micro pinches which can be explained with a radiative collapse model. The argon is not fully stripped of its electrons and is able radiate energy locally cooling the plasma. As the plasma cools, sausage instabilities form. When neon is used, a stable plasma column is produced. This regime has been discovered in other experiments. The stabilizing mechanism of the plasma is not understood. There may be a sheared flow stabilizing the column.

2.3 Z-pinches

The first Z-pinches were made in Holland by Martinus van Marum in 1790 [23]. He used one hundred Leyden jars to explode a one meter long wire. The implosion of the $\mathbf{J} \times \mathbf{B}$ was not postulated until 1905. The Bennett relation

$$8\pi NkT(1+Z) = \mu_0 I^2 \tag{2.4}$$

where N is the density per unit length, T is the temperature, and I is the current through the plasma, was derived in 1934. The declassification of the fusion program was triggered

by Kurchatov's lecture at Hartwell, where he identified an instability which leads to the neutron production seen in Z-pinches. In 1958 Anderson confirmed Kurchatov's results. Z-pinch work today is concentrating on obtaining powerful pulsed soft x-rays to driven inertial confinement fusion. They have also been used to produce neutrons, x-ray lasers, and study radiative collapse.

Z-pinches are created in a number of ways. The space between two electrodes may be filled with a uniform gas or frozen deuterium fibers. The onset of the Rayleigh-Taylor instability can be delayed by placing a large number of wires in a single or double array between the electrodes. A double shell puffed gas has been used to simulate a double wire array has also been studied [63]. Regardless of how the Z-pinch plasma is formed, it normally occupies a small volume, the initial Z-pinch radius is less than 4 cm and Z-pinch lengths are less than 4 cm. The large energy densities of these plasma is obtained by using capacitor banks with a quarter period of 200 ns to provide about 4 MA to the plasma.

In recent years, the wire array Z-pinch has been the most interesting because these Z-pinches are an efficient way to produce powerful short pulsed x-rays. A wire array consists of a large number (240) of thin (5 μm) wires. When the current is first applied each wire explodes while the entire array begins to implode. During this time there are different types of plasma, the coronal plasma can be modeled with the MHD equations, while the plasma in the core is in the resistive regime. The plasma from the individual wires merge together forming a conducting shell. The perturbation level of the $m = 0$ instability in the shell is $n^{-1/2}$ times the perturbation of level of a single wire where n is the number of wires. The $m = 0$ mode continues to grow while the shell implodes onto the axis. When the Z-pinch forms, most of the energy is in the kinetic energy of the ions. There is a rapid conversion to thermal energy followed by a slower equipartition to the electrons. The Z-pinch becomes unstable to the $m = 0$ mode shortly after this time.

2.4 $\mathbf{E} \times \mathbf{B}$ *Shear*

Transport barriers at the edge of tokamak plasma were first found in 1982 [68]. They have also been found in the core of tokamaks. Since this time worldwide experimental

and theoretical efforts have revealed new physics in the formation of these barriers. These bifurcation states are achieved by adding energy into the core of plasma. The magnetic and $\mathbf{E} \times \mathbf{B}$ flow shear increases, reducing the transport in the plasma. It is a rarity that the addition of energy to a system decreases the energy transport of the system.

It appears that flow shear breaks up islands responsible for anomalous transport [28]. Low frequency turbulence is first suppressed across the core of the plasma [15]. High frequency turbulence is then locally reduced in the internal transport barrier region. This region moves outward until $\gamma_{E\times B} \approx \gamma_{lin}^{max}$ where $\gamma_{E\times B}$ is the characteristic rate for shearing the turbulence and γ_{lin}^{max} is the fastest linear growth rate of the dominant instability [62]. In this region, particle and energy transport are reduced.

2.5 Sheared Flow Stabilization

The stability of a Z-pinch is one of the first configurations studied when teaching students linear stability. Theoretically it is easy to prove a static pure Z-pinch is unstable to the $m = 0$ and $m = 1$ mode. The results from coaxial guns, plasma focus, and $E \times B$ shear have shown the importance of plasma dynamics on the understanding and stability of the magnetic configurations. The stability requirements for a Z-pinch plasma should change when a flow is added. Stability analysis and computer simulations have shown that the $m = 0$ and $m = 1$ modes can be stabilized in a Z-pinch with a sheared axial flow [58].

Although the equilibrium of the Z-pinch, Eq. 1.6, is unaffected when an axial flow is added, the stability problem can no longer be solved analytically. The ideal MHD equations are linearized and combined to yield a set of coupled first order linear differential equations. The growth rates are then found with shooting codes. Equilibria which satisfy the Kadomtsev condition

$$-\frac{d\ln P}{d\ln r} \leq \frac{4\gamma}{2+\gamma\beta} \tag{2.5}$$

where γ is the ratio of specific heats and $\beta = 2\mu_0 P/B^2$, are used to stabilize the $m = 0$ mode. Shumlak and Hartman calculated the velocity shear needed to stabilize the kink mode as a function of r_{wall}/a, where r_{wall} is the radius of the wall and a is the characteristic radius of the pinch. When $r_{wall}/a = 4$, the required shear is no longer a function of r_{wall}/a.

A uniform velocity shear of

$$\frac{dV_z}{dr} \geq 0.1kV_A,$$

where V_z is the axial velocity, k is the wave number and V_A is the Alfvén velocity, would stabilize the kink mode [58]. Other authors have derived different sheared flow criteria which would stabilize the sausage and kink modes [1, 16, 51, 60].

Chapter 3

THE FORMATION PROCESS

Historically plasma accelerators have seen stable, pinch-like structures at the exit of the device. It has been postulated that the flow in this structure may be responsible for the stability. This observation leads to initiating a Z-pinch with an accelerator. This chapter gives an overview of how ZaP uses the results from previous experiments to study sheared flow stabilization. The project uses a coaxial accelerator coupled with an assembly region, shown in Fig. 3.1, which has never been done before. As seen in the previous chapter, there

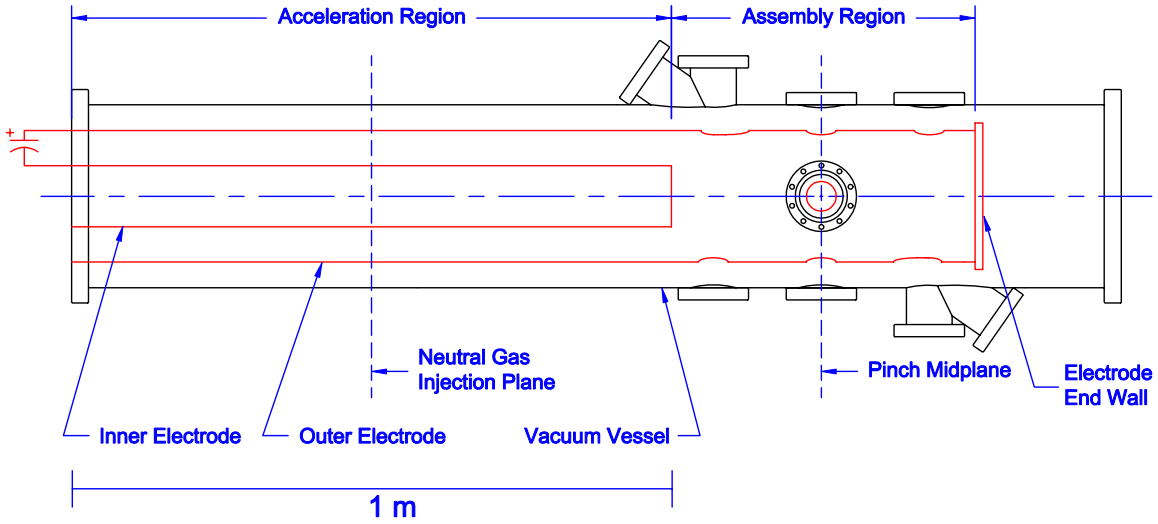

Figure 3.1: Diagram of the original configuration of ZaP experiment showing the major components of the experiment. The acceleration region is 1 m long with gas puff valves located at the midplane of the region. The inner electrode is a hollow pipe from the gas puff valves to the assembly region. The original assembly region was 0.5 m long with a solid endwall. The origin of coordinate system used in the experiment is placed at the midpoint of the original Z-pinch ($z = 0$ is located at the Z-pinch midplane).

is a large amount of research which has been done on plasma guns and static Z-pinches. A

focus of the experiment is studying the equilibrium and stability of a Z-pinch with a sheared axial velocity. This section relates the results from other devices to explain formation of the Z-pinch and identify processes in the accelerator important to the stability of the Z-pinch. Section 3.1 describes the formation process. Models of the current sheet rundown are discussed in Sec. 3.2.

3.1 Z-pinch Formation

ZaP uses a coaxial accelerator to form and accelerate the plasma, establishing a flow in the Z-pinch. A conceptual representation of producing a flow in a Z-pinch is given in Fig. 3.2. Gas is puffed at the midplane of the inner electrode. After a sufficient amount of time the voltage is applied to the electrodes. The gas breaks down, forming a uniform current sheet. The $\mathbf{J} \times \mathbf{B}$ force accelerates the current sheet down to the assembly region. In the accelerator, the leading edge of the current sheet forms a snowplow. The shape is given by $z \propto -r^2$ since the forces are greater near the inner electrode. The neutral gas in front of the current sheet is ionized in the shock. Although some of the plasma is entrained in the current sheet, most of the plasma is pushed up against the outer electrode by the snowplow. This is the description given by the LASL experiment for their coaxial gun studies and plasma foci.

After the plasma reaches the end of the acceleration region, a Z-Pinch is formed in the ZaP experiment. The LASL experiment let their plasma flow into a larger vacuum chamber. The outer electrode on ZaP extends beyond the inner electrode. The connection region of the current sheet along the inner electrode collapses on the axis at the end of the inner electrode, as shown in Fig. 3.3. The current sheet continues collapsing while the connection region along the outer electrode continues to travel along the outer electrode. The current sheet reaches an end wall where the connection region collapses onto the axis of the machine, forming a Z-pinch. It is difficult to imagine forming a Z-pinch with this process with out an embedded flow.

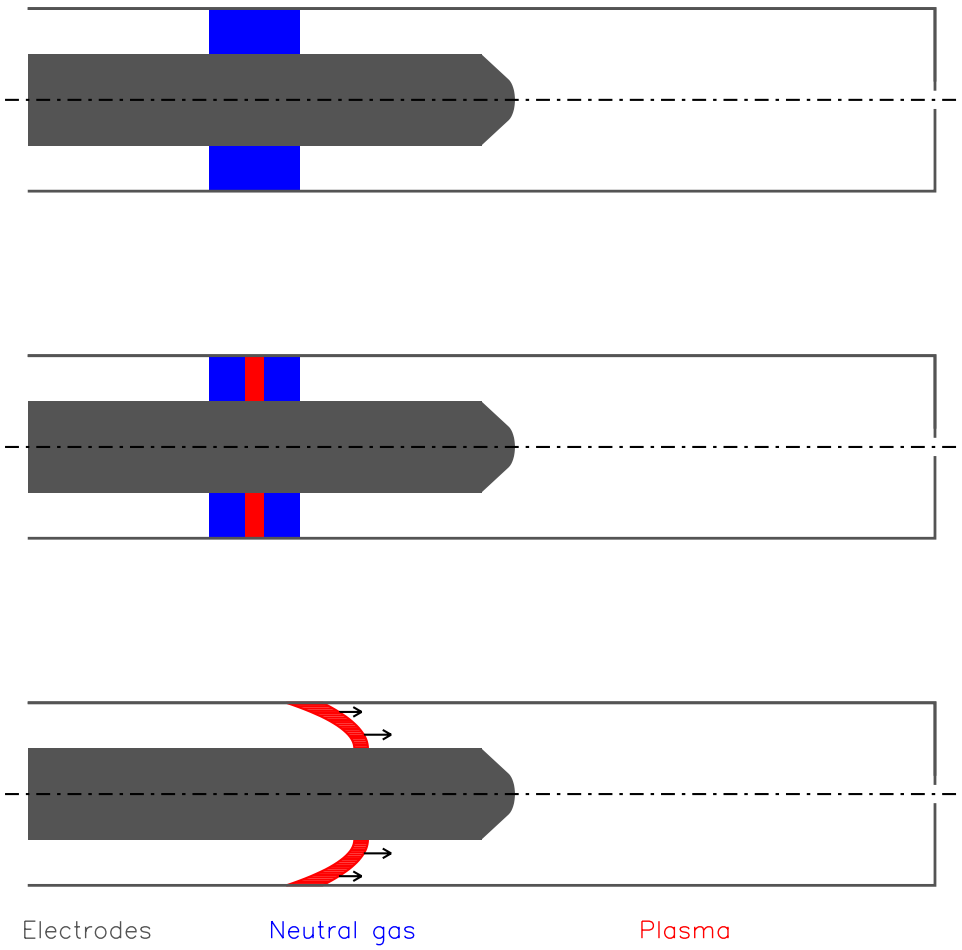

Figure 3.2: Conceptual drawing showing the initial steps in forming a sheared flow Z-pinch are similar to the LASL experiments. The dashed line is the center line of the experiment. The rest of the colors are described at the bottom of the frames. The first frame shows the distribution of neutral gas just before the voltage is applied. The next two frames show the breakdown of the gas after the voltage is applied and the initial rundown of the current sheet. The velocity of the plasma is shown as black arrows.

3.2 Snowplow Models

During the initial phase of the experiment the plasma should behave similarly to the LASL and plasma focus experiments. Two models may be used to describe the plasma in the accelerator when it is a snowplow. The first model, developed by Karpov *et al.* [30] uses

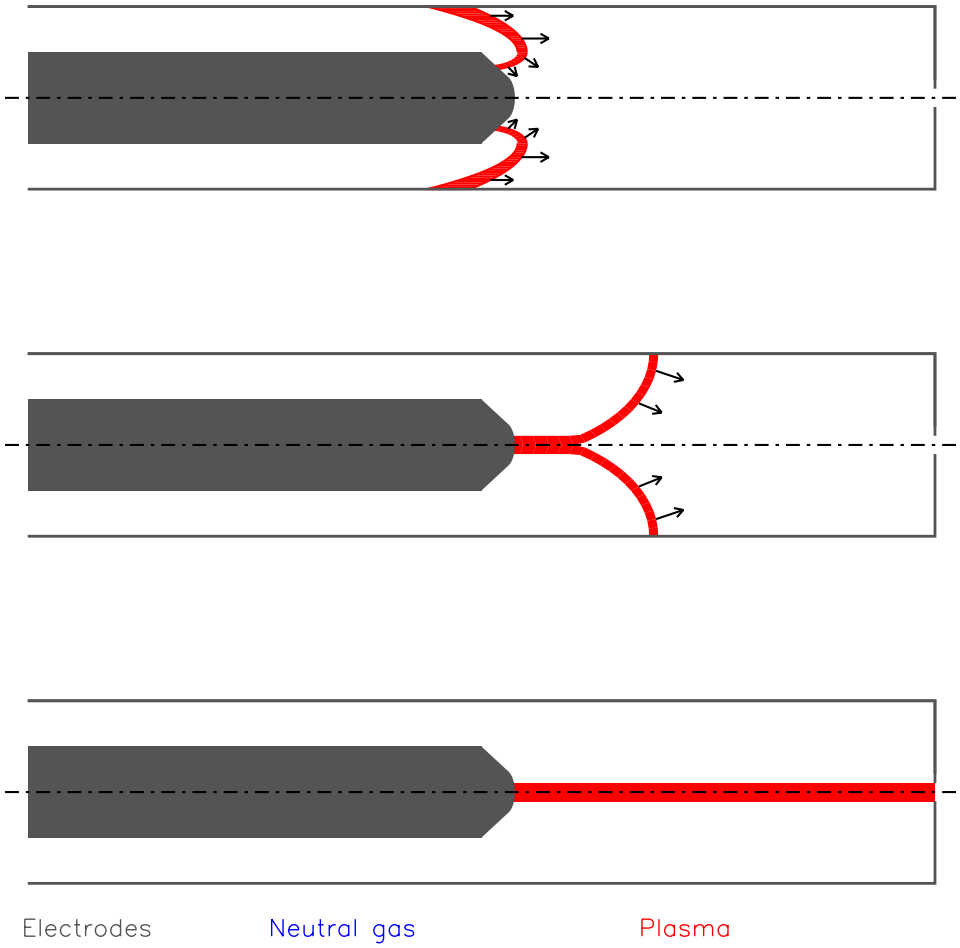

Figure 3.3: Conceptual drawing showing the final formation steps in ZaP are different than the LASL experiments. The dashed line is the center line of the experiment. The rest of the colors are described at the bottom of the frames. The first two frames show the current sheet collapsing onto the axis of the machine as the current sheet travels along the outer electrode. The final frame shows the current sheet after it has completely collapsed and formed a Z-pinch. The velocity of the plasma is shown as black arrows.

the circuit parameters and the fill pressure to describe the total current and location of the current sheet as functions of time. The second model, developed at LASL by Butler *et al.*, describes the shape of the current sheet when the current is constant. These two models are described in following two sections. Comparison of the models and the initial phases of the ZaP experiment will be discussed in Sec. 7.1.

3.2.1 Varying Current Snowplow

A snowplow model, where the current varies with time, was developed by Karpov *et al.* [30]. The modified equation of motion

$$\frac{d}{dt}\left(z\dot{z}\right)+\frac{\gamma-1}{2}\dot{z}^2=\frac{\mu_0}{2k\rho}\left(\frac{I}{2\pi r}\right)^2 \tag{3.1}$$

where ρ is the density of the neutral gas, γ is the ratio of specific heats, $k=k_i/\beta_i=$ constant, k_i is the plowing efficiency, and β_i is the decrease of magnetic pressure due to the curvature; and Kirchhoff's equation

$$\frac{d}{dt}\left[\left(L_0+L(t)\right)I\right]+R_0I=V_0-\frac{1}{C_0}\int Idt \tag{3.2}$$

where L_0, R_0, and C_0 are the inductance, resistance and capacitance of the driving circuit, are used to describe the motion of the snowplow. $L(t)$ is the change in the inductance as the current sheet travels along the annulus. Separation of variables can be used when the shape of the snowplow is given by $z(r,t)=r_0z_1(r_0,t)/r=r_0z_1(t)/r$, where r_0 is the radius of the inner electrode. The variable inductance becomes

$$L(t)=\frac{\mu_0}{4\pi}bz_1(t)$$

where $b=2\left(r_1-r_0\right)/r_1$ and r_1 is the radius of the outer electrode. Equations 3.1 and 3.2 are then rewritten with the nondimensional parameters

$$\tau=\frac{t}{\sqrt{L_0C_0}}\ ,y=z_1\frac{\mu_0}{4\pi}\frac{b}{L_0}\ ,\varphi=I\frac{\sqrt{L_0C_0}}{V_0C_0}$$

to form

$$\frac{1}{2}\frac{d^2}{d\tau^2}y^2+\frac{\gamma-1}{2}\left(\frac{dy}{d\tau}\right)^2=q\varphi^2 \tag{3.3}$$

$$\frac{d^2}{d\tau^2}\left[\left(1+y\right)\varphi\right]+p\frac{d\varphi}{d\tau}+\varphi=0 \tag{3.4}$$

where

$$q = \frac{C_0^2V_0^2b^2}{2\pi r_0^2\rho L_0^2k}\left(\frac{\mu_0}{4\pi}\right)^3 \qquad \text{and}$$

$$p = R_0\sqrt{\frac{C_0}{L_0}}.$$

These equations are numerically integrated with the initial conditions

$$y(0) = \dot{y}(0) = \varphi(0) = 0$$
$$\dot{\varphi}(0) = 1$$

using the second order accurate finite-difference operators

$$y(\tau_n) = y_n \tag{3.5}$$
$$\varphi(\tau_n) = \varphi_n \tag{3.6}$$
$$\dot{y}(\tau_n) = \frac{1}{2\triangle\tau}\left(y_{n+1} - y_{n-1}\right) \tag{3.7}$$
$$\dot{\varphi}(\tau_n) = \frac{1}{2\triangle\tau}\left(\varphi_{n+1} - \varphi_{n-1}\right) \tag{3.8}$$
$$\ddot{y}(\tau_n) = \frac{1}{\triangle\tau^2}\left(y_{n+1} - 2y_n + y_{n-1}\right) \tag{3.9}$$
$$\ddot{\varphi}(\tau_n) = \frac{1}{\triangle\tau^2}\left(\varphi_{n+1} - 2\varphi_n + \varphi_{n-1}\right). \tag{3.10}$$

Substituting Eqs. 3.5 to 3.10 into Eqs. 3.3 and 3.4 and solving for y_{n+1} and φ_{n+1}, the following equations are obtained

$$y_{n+1} = \frac{-b_y + \sqrt{b_y^2 - 4a_y c_y}}{2a_y} \tag{3.11}$$
$$\varphi_{n+1} = \frac{\left[2\left(1 + y_n\right) - \triangle\tau^2\right]\varphi_n + \left[\frac{P}{2}\triangle\tau - \left(1 + y_{n-1}\right)\right]\varphi_{n-1}}{1 + y_{n+1} + \frac{P}{2}\triangle\tau} \tag{3.12}$$

where

$$a_y = 2 + \frac{\gamma - 1}{2}$$
$$b_y = -(\gamma - 1)y_{n-1}$$
$$c_y = -4y_n^2 + \frac{\gamma + 3}{2}y_{n-1}^2 - 4\triangle x^2 q\varphi_n$$

An IDL routine, *snow_plow_time.pro* Sec. C.1.1, can be used to compare the location of the current sheet and the total current to experimental data.

3.2.2 Snowplow Shape

The shape of the snowplow during a constant current discharge was described by Butler *et al.* [9, 10]. In the shock frame the stationary equations of the shock layer for force balance,

conservation of mass, momentum and energy are

$$0 = \frac{B_0^2 r_0^2}{2\mu_0 r^2} - \rho u^2 \cos^2\theta + \frac{\sigma V^2}{2\pi r}\frac{d\theta}{ds} \tag{3.13}$$

$$\frac{d}{ds}(V\sigma) = 2\pi r u \rho \cos\theta \tag{3.14}$$

$$\frac{d}{ds}\left(V^2\sigma\right) = 2\pi r u^2 \rho \cos\theta \sin\theta - 2\pi r \delta \frac{dP}{ds} \tag{3.15}$$

$$\frac{d}{ds}(V\varepsilon) = \frac{2\pi r u}{\gamma}\frac{\rho}{2}\cos\theta\left[u^2\cos^2\theta + (u\sin\theta - V)^2 - u_c^2\right] + \frac{2\pi r\delta}{\gamma}\frac{dP}{ds} \tag{3.16}$$

where s is the arc length along the shock from the electrode, r_0 is the electrode radius, B_0 is the magnetic field at the electrode surface behind the current sheet, ρ is the mass density of the gas ahead of the shock, u is the flow velocity of the undisturbed gas, θ is the angle between the shock surface and the local radius, V is the velocity in the shocked layer along the shock, σ is the mass per unit arc length (integrated around the center axis), ε is the internal energy per unit arc length, u_c is the velocity at which the kinetic energy of incoming molecules equals the ionization energy, γ is the ratio of specific heats, δ is the thickness of the shocked layer, and P is the pressure in the shock layer. The pressure in the shocked layer is taken as the average of the pressure in the front and the back of the shock

$$P = \frac{B_0^2 r_0^2}{2\mu_0 r^2} + \rho u^2 \cos^2\theta \ . \tag{3.17}$$

The internal energy in the shock layer is related to the pressure by

$$P = nkT = \frac{(\gamma - 1)\varepsilon}{2\pi r \delta} \tag{3.18}$$

where $\varepsilon/2\pi r\delta$ is the internal energy per unit volume. The radial and axial position of the shock can be found with

$$\frac{dr}{ds} = \cos\theta \tag{3.19}$$

$$\frac{dz}{ds} = -\sin\theta \tag{3.20}$$

This system of equations can be solved numerically starting at the singular point $s = 0$, where $\delta = 0$, $v = 0$, and $\sigma = 0$.

Before solving the system of equations, Eq. 3.13 to Eq. 3.16 are written in non-dimensional form. The snowplow has free slip boundary condition at the inner electrode, $\theta = 0$ and $d\theta/ds = 0$. A consequence is

$$\frac{B_0}{2\mu_0} = \rho u^2. \tag{3.21}$$

The following nondimensional parameters

$$\begin{aligned}
s_n &= \frac{s}{r_0} \quad , \quad r_n = \frac{r}{r_0} \\
V_n &= \frac{V}{u} \quad , \quad \sigma_n = \frac{\sigma}{2\pi r_0^2 \rho} \\
z_n &= \frac{z}{r_0} \quad , \quad P_n = \frac{P}{\frac{1}{2}\rho u^2} \\
\delta_n &= \frac{\delta}{r_0} \quad , \quad \varepsilon_n = \frac{\gamma\varepsilon}{2\pi r_0^2 \frac{1}{2}\rho u^2} \\
u_{c\ n} &= \frac{u_c}{u}
\end{aligned}$$

can then be used along with Eq. 3.21 to rewrite Eq. 3.13 to Eq. 3.16 as

$$\frac{d\theta}{ds_n} = \frac{r_n}{\sigma_n V_n^2}\left(\cos^2\theta - \frac{1}{r_n^2}\right) \tag{3.22}$$

$$\frac{d}{ds_n}\left(\sigma_n V_n\right) = r_n \cos\theta \tag{3.23}$$

$$\frac{d}{ds_n}\left(\sigma_n V_n^2\right) = r_n \cos\theta \sin\theta - \frac{r_n \delta_n}{2}\frac{dP_n}{ds_n} \tag{3.24}$$

$$\begin{aligned}
\frac{d}{ds_n}\left(\varepsilon_n V_n\right) &= r_n \cos\theta \left[\cos^2\theta + (sin\theta - V_n)^2 - (u_{c\ n})^2\right] + \\
&\quad r_n \delta_n \frac{dP_n}{ds_n}
\end{aligned} \tag{3.25}$$

$$P_n = \frac{1}{r_n^2} + \cos^2\theta \tag{3.26}$$

$$\frac{dP_n}{ds_n} = -2\cos\theta\left(\frac{1}{r_n^3} + \sin\theta\frac{d\theta}{ds_n}\right) \tag{3.27}$$

$$\delta_n = \frac{\gamma - 1}{\gamma}\frac{\varepsilon_n}{r_n P_n} \tag{3.28}$$

When the pressure, internal energy and the centrifugal force terms are disregarded in

Eqs. 3.22 through 3.28, the remaining set of equations

$$\begin{aligned} \cos^2\theta &= \frac{1}{r_n^2} \\ \frac{d}{ds_n}\left(\sigma_n V_n\right) &= r_n \cos\theta \\ \frac{d}{ds_n}\left(\sigma_n V^2\right) &= r_n \cos\theta \sin\theta \end{aligned}$$

have an analytic solution.

$$r_n = \sqrt{1+2s_n} \tag{3.29}$$

$$z_n = \frac{1}{2}\left(\sqrt{s_n\left(2+4s_n\right)} - \operatorname{arcsinh}\sqrt{2s_n}\right) \tag{3.30}$$

$$\sigma_n = 2s_n^2\left(\sqrt{s_n\left(2+4s_n\right)} - \operatorname{arcsinh}\sqrt{2s_n}\right)^{-1} \tag{3.31}$$

$$V_n = \frac{1}{2s_n}\left(\sqrt{s_n\left(2+4s_n\right)} - \operatorname{arcsinh}\sqrt{2s_n}\right) \tag{3.32}$$

While this set of equations describes the shape of the snow plow, the thickness and density of the snowplow are lost.

The full set of equations are solved by expanding about the singular point at $s = 0$, and using the leading terms to start the numeric integration scheme. The equations are expanded about $s = 0$, where $\delta = 0$, $V = 0$, $\sigma = 0$, and $\theta = 0$, keeping only the leading terms which are proportional to $s_n^{1/2}$. The following forms of solutions are assumed for θ, σ_n, V_n, ε_n, and r_n

$$\theta = a_\theta s_n^{1/2} \tag{3.33}$$

$$\sigma_n = a_\sigma s_n^{1/2} \tag{3.34}$$

$$V_n = a_V s_n^{1/2} \tag{3.35}$$

$$\varepsilon_n = a_\varepsilon s_n^{1/2} \tag{3.36}$$

$$\cos\theta = 1 \tag{3.37}$$

$$r_n = 1 + s_n \tag{3.38}$$

The constants are found by substituting these solutions into Eqs. 3.22 to 3.28 and ignoring

higher order terms since $s \approx 0$. After some algebra the constants become

$$a_\theta = \frac{2}{\sqrt{\frac{8}{3} + \frac{\gamma-1}{3\gamma}\left(1 - u_{c\ n}^2\right)}} \tag{3.39}$$

$$a_\sigma = \frac{1}{\sqrt{\frac{8}{3} + \frac{\gamma-1}{3\gamma}\left(1 - u_{c\ n}^2\right)}} \tag{3.40}$$

$$a_V = \sqrt{\frac{8}{3} + \frac{\gamma-1}{3\gamma}\left(1 - u_{c\ n}^2\right)} \tag{3.41}$$

$$a_\varepsilon = \frac{1 - u_{c\ n}^2}{\sqrt{\frac{8}{3} + \frac{\gamma-1}{3\gamma}\left(1 - u_{c\ n}^2\right)}} \tag{3.42}$$

This gives the first point above inner electrode. Once the parameters at this point are found, a Runge-Kutta numeric integration scheme is used to determine the shape of the snowplow. The shape of a snowplow is shown in Fig. 3.4, calculated with the IDL routine, *con_cur_sp_dis.pro*, Sec. C.1.2. The largest effect of changing the parameters is seen by changing $u_{c\ n}$. The thickness of the current sheet approaches 0 as $u_{c\ n}$ approaches 1. All of the incoming particle's kinetic energy is used to ionize the neutral gas resulting in a cold, thin shocked region. ZaP does not presently have the capability to verify the shape of the snowplow. The results are useful when comparing the time evolution of various diagnostics in the acceleration region.

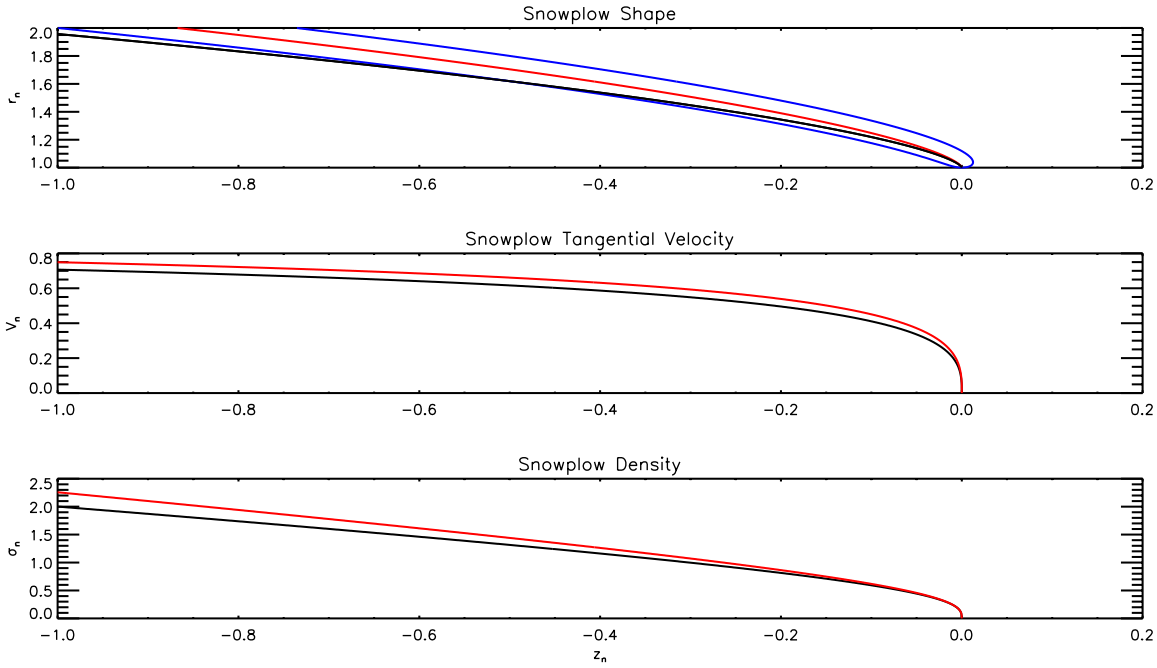

Figure 3.4: The shape and parameters of the snowplow are found using the analytic expression (black) and numeric integration (red). The blue curve in the top graph shows the thickness of the snow plow.

Chapter 4

DESIGN OF THE ZAP EXPERIMENT

The ZaP experiment is interested in studying fusion grade plasmas. The geometry of the machine is set by the formation process described in the last chapter. The dimensions of the electrodes have been chosen to be similar to the LASL devices, because of the similarity to the initial stages of the formation process and the pinch-like structures which were seen. Improvements in the design of the systems used on the ZaP experiment have been made with the technology avaliable. The major components of the machine are shown in Fig. 4.1.

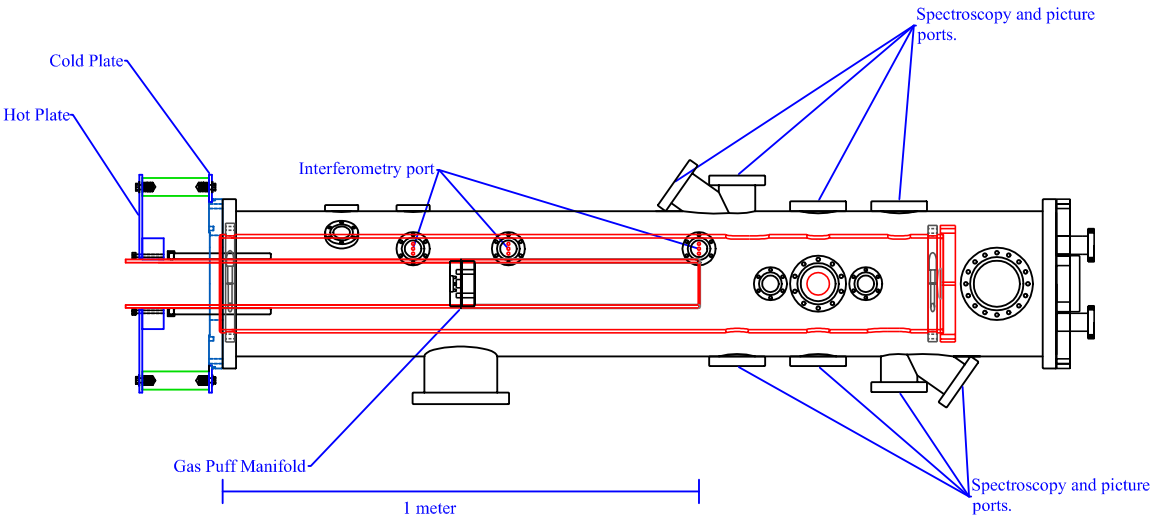

Figure 4.1: Diagram of the original ZaP vacuum chamber showing the major components of the tank with the end flanges attached. The diagnostic ports on the top and bottom of the machine are labeled. The large port in the assembly region is used for holography, interferometry, and optical images of the Z-pinch. The two smaller ports on either side of this port are used for internal probes. The port located to the right of the endwall is the electrical feed through port. The unlabeled ports on the left side of the machine are used for the vacuum system.

The design of the machine must allow for diagnostic access. Holes drilled in the electrodes have a minimal impact on the rundown of the current sheet and formation of the Z-pinch. Many of the planned diagnostics required multiple plasma pulses to conduct a survey. Systems used in the experiment are reproducible, *i.e.* constant gas fills, little jitter on the ignitron timing, *etc.* The down time between pulses has been kept to a few minutes to ensure that surveys can be completed in a day. The rapid cycling of the experiment puts an added pressure on the hardware. This goal has guided the design of most of the systems.

This chapter begins by describing the vacuum system. A vacuum of 10^{-8} torr has been created in the vacuum chamber. The vacuum tank design also allows for diagnostic access to the plasma. An interlock system on the vacuum system is used to protect the system from operator error. Inside the chamber the electrodes supply the current to the plasma and withstand interruptions of the Z-pinch without failing. The gas puff valves, which inject neutral gas into the annulus are described in Sec. 4.2. Section 4.3 describes the multiple capacitor bank configurations have been used to supply currents. A circuit diagram of the ZaP experiment has found to model currents and voltages. Safety interlocks in the lab prevent employees from being near the experiment while pulses are being taken are discussed in Sec. 4.4. As the understanding of the experiment has increased many of these systems have been modified and improved. Figure 4.2 shows the present configuration of the ZaP vacuum tank and electrodes. The original design and changes made are described in each section.

4.1 Vacuum System

The vacuum system consists of the vacuum tank, pumping system, electrodes and diagnostics inside the vessel. All of these parts are integrated together to achieve a low base pressure of 10^{-8} torr. An interlock system is used to control the vacuum pumps and valves.

4.1.1 Creating a Low Ultimate Pressure

Most vacuum systems are able to generate a high vacuum with a roughing pump and a turbomolecular pump. Normally a rough pump is used to create a vacuum of 10^{-3} torr

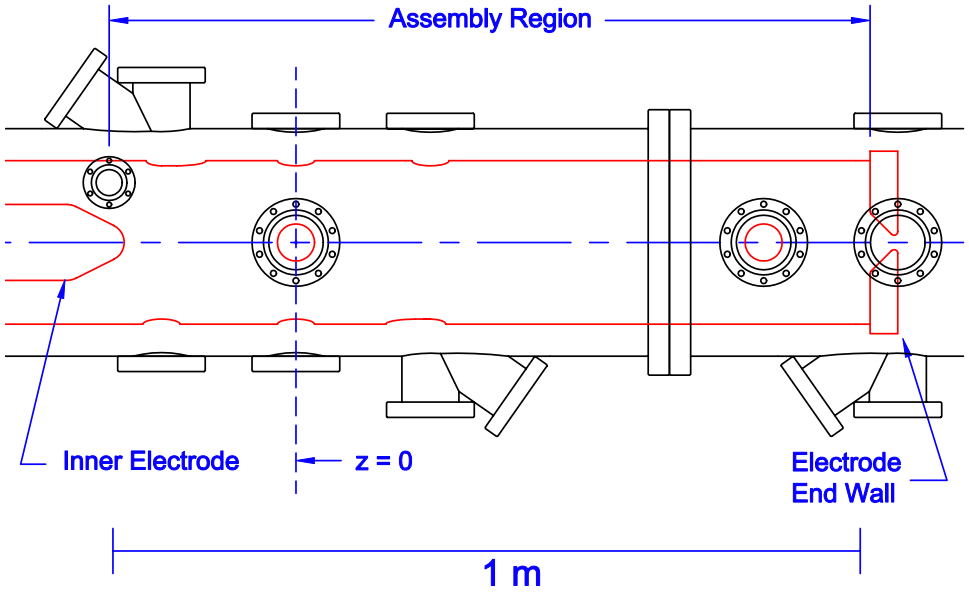

Figure 4.2: Diagram of the present ZaP vacuum chamber showing the changes made to the electrodes and vacuum chamber. The hollow inner electrode has been replaced with a nose cone. The assembly region has been lengthened to 1.0 m with a 0.5 m outer electrode extension. A hole has been added to the end wall to remove a possible stagnation point in the plasma. A 0.75 m long extension has been added to the vacuum chamber.

that the turbo pump can operate in. The turbo pump then creates a high vacuum in the chamber. In ZaP, the vacuum is created in the tank with three pumps. A Leybold D60A mechanical pump is used create a vacuum in the rough lines. The pump does not have throughput needed for helium glow discharge cleaning in the chamber. A Leybold WA251 roots blower has been added to increase the throughput of the roughing system. The addition of the roots blower has decreased the ultimate pressure of the rough lines to 10^{-4} torr. A Leybold TMP450 Turbovac then creates the vacuum in the chamber. The lowest base pressure obtained in ZaP was 2×10^{-8} torr.

Since gas is injected into the chamber each pulse, the pumping system must be able to achieve the high vacuum in a short time. The pump down time was kept as short as possible by keeping the conductance of the vacuum system much larger than the pumping speed of the turbo pump. Four inch diameter copper tubing was used for the rough lines to keep the

conductance of the roughing system larger than the pumping speed of the roughing pumps.

Low ultimate pressures, of 10^{-8} Torr, are achieved by using nearly all metal seals on the chamber. All of the ports on the chamber, including the tank end plates, use copper gasket conflat seals. All of the windows, electrical feedthroughs, gate valves, and pressure gauges are welded on to conflate blanks. Most of these are standard parts which can be purchased from various venders. The electrical feedthroughs consisting of three DB25 electrical feedthroughs are mounted on a 6 inch conflats were a special order item from ISI. The only seals which do not use conflates are around the Alumina insulator. These seals are made with Viton O-rings. The O-Rings are held in place with stainless steel pushers. Every seal was checked with a helium leak checker.

Even in a completely sealed chamber, virtual leaks and outgasing can keep the pressure high. Virtual leaks are caused by trapped volumes in the vacuum. The trapped volume can be the empty space behind a bolt or space behind a weld. Air trapped inside this volume slowly escapes into the vacuum, keeping the pressure high without showing up on a leak check. To avoid virtual leaks, there must not be any trapped volumes inside the vacuum chamber. All of the welds are made on the inside of the chamber to avoid virtual leaks from the welds. To prevent a trapped volume from being created in the space between a bolt and a threaded hole, the bolts either have a center hole or they have the threads on one side ground down. Materials in vacuum can also outgas. Contaminants which adhere to material surfaces slowly escape into the vacuum acting like a leak. Most of the materials used in the vacuum chamber were chosen to avoid this problem. Without virtual leaks and outgasing in the chamber, a high vacuum can be obtained.

When changes are made inside of the vacuum chamber, water and other impurities can contaminate the surfaces in the chamber. Over pressurizing the chamber with dry nitrogen while the port covers are removed will decrease the contamination of the surfaces. As the contaminants sublimate, they will eventually be pumped out of the system. This process is enhanced when the temperature of the tank is elevated. All of the materials in the ZaP vacuum chamber are chosen such the tank temperature can be raised to 200 C with heat tapes. Heat lamps and hot air guns are used to raise the temperature of components which can't be directly heated with heat tape. In the future, the electrode surfaces may be cleaned

with He glow discharge cleaning.

4.1.2 Vacuum Interlock System

An interlock system is used on the vacuum system to prevent possible damage of the vacuum hardware if the pumps are started out of sequence, or the wrong valve is opened. The vacuum system controls when a pump can be started or a valve can be opened as shown in the logic diagram, Fig. 4.3. This system can be bypassed by an experienced operator when a potentially unsafe operation is needed. Additional protections are needed for the roots blower, shown in Fig. 4.4. The most expensive pump on the system is the turbo pump. This pump has the most safeguards as shown in Fig. 4.5. Most of the safeguards for this pump cannot be overridden. The vacuum system runs 24 hours a day, seven days a week. The interlock system protects vacuum components when no one is working in the lab. The system shuts down the pumps and closes all of the valves whenever there has been a power failure or the pressure in the roughing lines rises above 1 torr.

4.1.3 Vacuum Chamber

The vacuum chamber consists of a vacuum tank, endplates, and all of the diagnostic port covers. The main vacuum tank is a 69 inch long, 304 stainless steel pipe with a 12 inch diameter and a 0.187 inch wall thickness, shown in Fig. 4.1. This pipe is able to support a one atmosphere pressure difference without deforming or degrading the vacuum. Half nipples welded onto the pipe provide access for the vacuum and diagnostic systems. The ports closest to the hot plate are used to pump out the chamber, to measure the pressure and to supply dry nitrogen to the chamber. The three optical ports in the acceleration region are used for interferometry. The windows on these ports are made of Pyrex. The set of coaxial ports located on each side of the vacuum tank in the assembly region are covered with Pyrex windows. Fused silica windows are presently located on these ports because they can handle more power and be cleaned. These ports are used for interferometry and holography. Two smaller ports, located on each side of the holography port on the left side of the machine, are used for internal probes. The ports located on the top and bottom of the

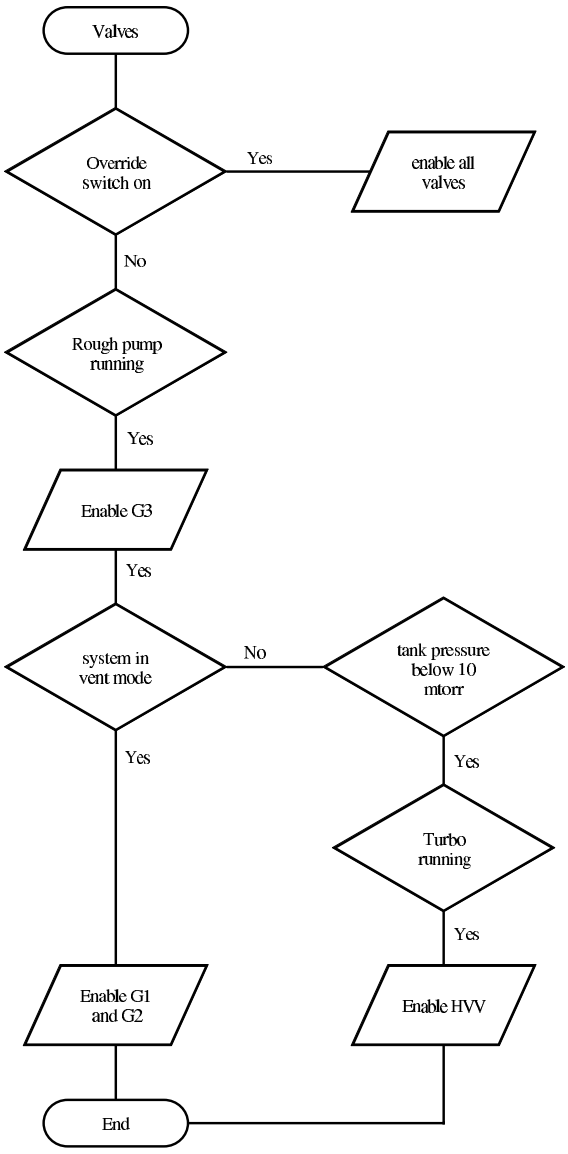

Figure 4.3: Flow chart of the logic used to design the vacuum interlocks for the valves on the vacuum system. Valves G1 and G2 connect the cold trap to the rough line and the vacuum tank. The turbo pump is connected to the rough lines through G3 and to the vacuum chamber through HVV. The valves are normally disabled.

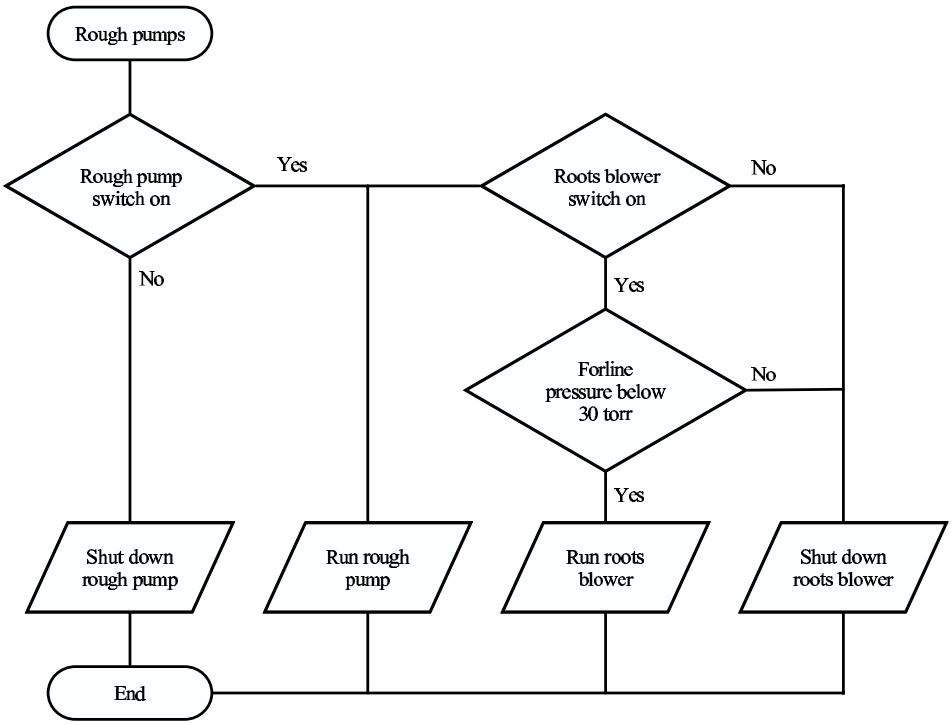

Figure 4.4: Flow chart of the logic used to design the vacuum interlocks for the roughing pumps.

tank are used for spectroscopy. These ports are covered with UV grade fused silica windows so that measurements of the C V triplet at 227.7 nm can be made. Two 6 inch nipples welded onto the side of the tank beyond the assembly region are used for the electrical feedthroughs. An extension was added to the chamber on August 20, 2002 so that the outer electrode could be extended. The extension is a 35.5 inch long, 304 stainless steel pipe with a 12 inch diameter and a 0.187 inch wall thickness, shown in Fig. 4.2. The ports on the extension serve the same functions as ports in assembly region of the main chamber.

The vacuum endplate at the assembly end of the tank has three optical access ports. These ports are used to study the breakdown of the plasma when the outer electrode endwall is removed. The vacuum endplate at the rundown end of the tank serves multiple purposes. The endplate serves as the cold plate. The outer electrode and hot plate are supported by this endplate. The inner electrode is insulated from the cold plate with an alumina

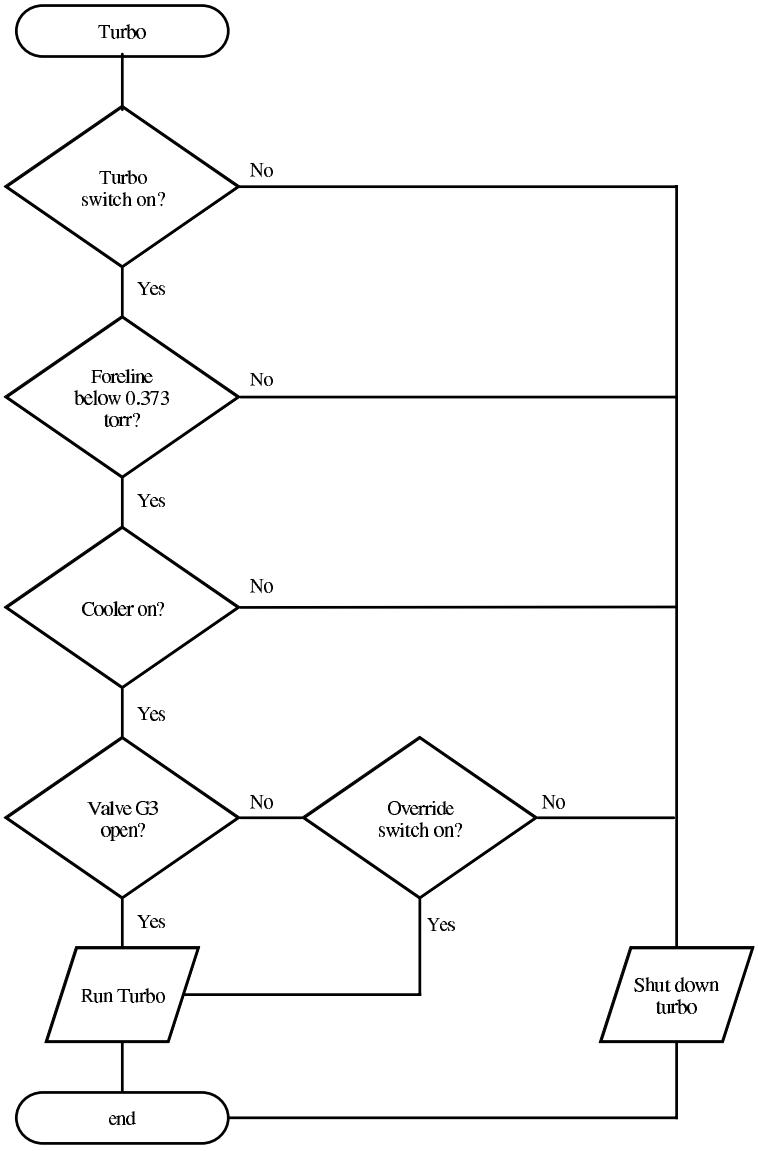

Figure 4.5: Flow chart of the logic used to design the vacuum interlocks for the turbo pump.

sleeve. The o-ring pusher for the fat Viton o-ring around the outside of the insulator is also mounted on this endplate.

Inner Electrode

The inner electrode is a 4 inch outer diameter, 0.25 inch wall thickness copper pipe. This size of pipe will not deform under the magnetic pressure nor will it sag appreciably when 4 feet of the pipe are cantilevered. The inner electrode is made of two pieces so that the shape of the electrode in the acceleration region can be changed. The original electrode in the acceleration region is an open-ended pipe with the assembly region end rounded. The inner electrode gas manifold is located at the connection of the two pieces. The section in the rundown region is sprayed with 0.010" of tungsten on the outer and inner surface. The other piece was not sprayed because blisters along the electrode formed when it was brazed. The blisters may have formed leaks along the electrode during the tungsten spraying process. The inner electrode is connected to the hot plate with a brass split ring. The electrode was centered by adding brass shims between the cold plate and G-10 spacers.

On August 17, 2002 the hollow inner electrode was replaced with a copper pipe with a nose cone. The pipe is from the piece used to make the original inner electrode and is tungsten sprayed. The end of the copper pipe has a threaded stainless steel insert which allows for variable nose cone shapes to be tested. The present nose cone has a 30° angle between the z axis and the face of the nose cone and a 1 inch radius at the z axis. Even though the nose cone is tungsten sprayed, ablation at the Z-pinch contact point has been a problem. A new design which uses graphite should mitigate this problem.

Outer Electrode

The outer electrode is made out of 7.583 inch inner diameter, 0.271 inch wall thickness copper pipe. The magnetic pressure will not deform this pipe. Holes are machined for the magnetic probes, outer gas puff, and optical access. The assemble region end of the pipe is supported with electrical insulating stands. A endwall electrode is attached to a stainless steel split ring at the assembly end of the electrode. A 500 kA current, uniformly distributed in a 2 cm diameter Z-pinch, will generate 9 tons of force on each pulse. The stresses in the endwall were calculated with ANSYS. When a static load was applied to the endwall, the weakest point of the design was the bolts. The shear stresses of the endwall and bolts are

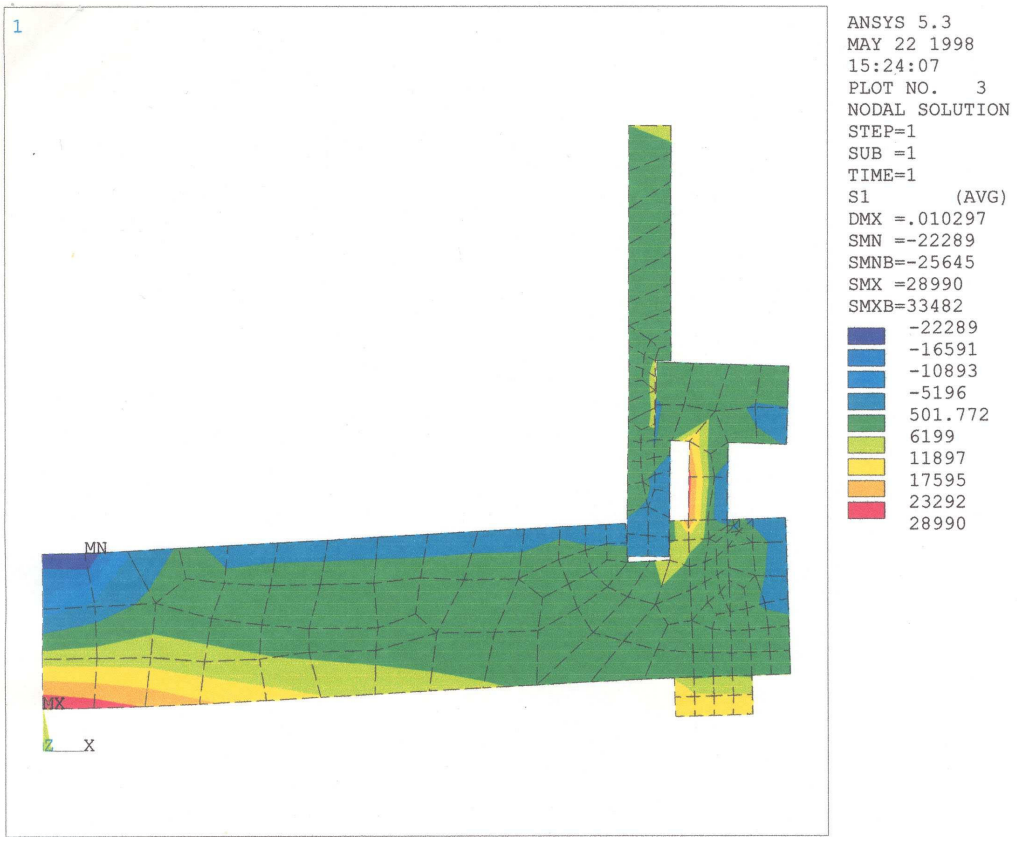

Figure 4.6: ANSYS was used to do the stress analysis of the endwall. The highest shear stresses (red) are on the bottom of the endwall and on the inner edge of the bolts. The thickness of the endwall is chosen such that the bolts will not deform when a static load is applied. The deflection is exaggerated. The contours are the shear stress in psi.

shown in Fig. 4.6. Increasing the thickness of the electrode, prevented the deformation of the bolts in the simulation. The simulation showed that the bolts did not deform when a dynamic loading was applied to the endwall electrode. A tungsten alloy insert is installed at the Z-pinch attachment location on the end wall electrode. A hole was drilled behind the insert to remove the trapped volume behind the insert. The inner surface of the outer electrode and the plasma-facing surface of the endwall electrode were sprayed with 0.010 inches of tungsten to prevent sputtering.

Two changes have been made to outer electrode and endwall. On August 17, 2002, the

endwall was replaced with a similar endwall with a hole in the middle. The hole allows plasma to exit the Z-pinch, avoiding a stagnation point which may have been terminating the velocity shear. The hole is 3.2 inches on assembly side of the endwall and tapers down to 0.8 inches on the endflange side of the endwall with rounded edges. On March 10, 2003, a 20 inch extension was added to the assembly region of the outer electrode lengthening the Z-pinch by 0.5 m. Holes are machined in the electrode for magnetic probes and optical access. Both of these components have been sprayed with 0.010 inches of tungsten on all plasma-facing surfaces to prevent sputtering.

4.2 Gas System

The hydrogen is puffed into the annulus between the electrodes with three fast valves. Deuterium is not used in ZaP because the 14 MeV neutrons which would be produced would restrict the access to the experiment. Bottled H_2 is purified with an Resource Systems, Inc. Hydrogen purifier. The maximum operating pressure of the purifier is about 150 psi which limits the pressure of the gas system. The hydrogen is then fed into a gas board where the line pressure is monitored. The gas board also allows for other gases to be used in the experiment. Stainless stain and Teflon tubing is used to supply the valves with the gas. Replacing the purifier with high pressure filters and Teflon tubing with a TFE 060 tube would allow the gas pressure to be increased to 1000 psi.

Three fast solenoid valves are used to inject the gas into the gas distribution manifolds. The inner electrode gas puff valve injects the gas into a 0.816 in^3 plenum. The gas is then injected into the annulus through eight equally spaced radial slots. One of the slots is located at top dead center of the inner electrode. Two slower gas puff valves are used on the outer electrode. Each valve injects the gas into a stainless steel tubing manifold which is inserted into four holes of the eight gas injection holes in the outer electrode. The length in the tubing between each hole and the valve is the same in insure the gas is injected at the same time. Each valve injects gas into every other hole, one valve injects gas into the top, bottom, and side holes while the other valve injects gas into the diagonal holes, uniformly filling the annulus with hydrogen. The two slower valves were replaced with eight valves

on May 2, 2002. The valves are identical to that used for the inner gas puff. These valves inject gas directly into the eight holes located on the outer electrode. Two valves which are opposite sides of the outer electrode are triggered with the same circuit. Better control of the gas fill has been achieved with these valves.

All of these valves are triggered with circuits designed and built at the University of Washington. The valve is over driven with 300 volts to increase its response time. Fiber optics are used to trigger the circuit and rechargeable batteries are used to supply the power. This is essential for the inner gas puff valve where the ground of the valve raises to the inner electrode potential on each shot.

The original gas puff characterization related the dial setting of the valves to the total gas injected into the vacuum chamber. The high vacuum valve was closed and the valves were puffed. The pressure of the tank was recorded after a few minutes. This process was repeated a number of times to improve the statistics. The time evolution of the gas pressure injected by the valves was characterized by T. Shreve [55]. With the two calibrations, the average neutral density in the accelerator is known. The time evolution calibration is used to ensure that gas is still being injected into the accelerator during the plasma pulse. This inhibits material from being liberated from the walls to supply the plasma with the need charge carriers.

4.3 Energy Storage

The energy required for each pulse is stored in a capacitor bank. The bank consists of groups of four or eight capacitors which are connected in parallel. Multiple groups can be connected in parallel to the experiment to increase the stored energy. Six different bank configurations have been used to supply the current to the experiment. The waveforms are shown in this section.

ZaP uses 170 μF capacitors with an maximum charge of 10 kV and an inductance of 0.04 μH. Brass bars are used to connect the capacitors together. The voltage to the hot plate is switched with D-size ignitrons mounted in brass tubes to reduce the inductance and the noise from the switching process and reduce electromagnetic noise. When the maximum

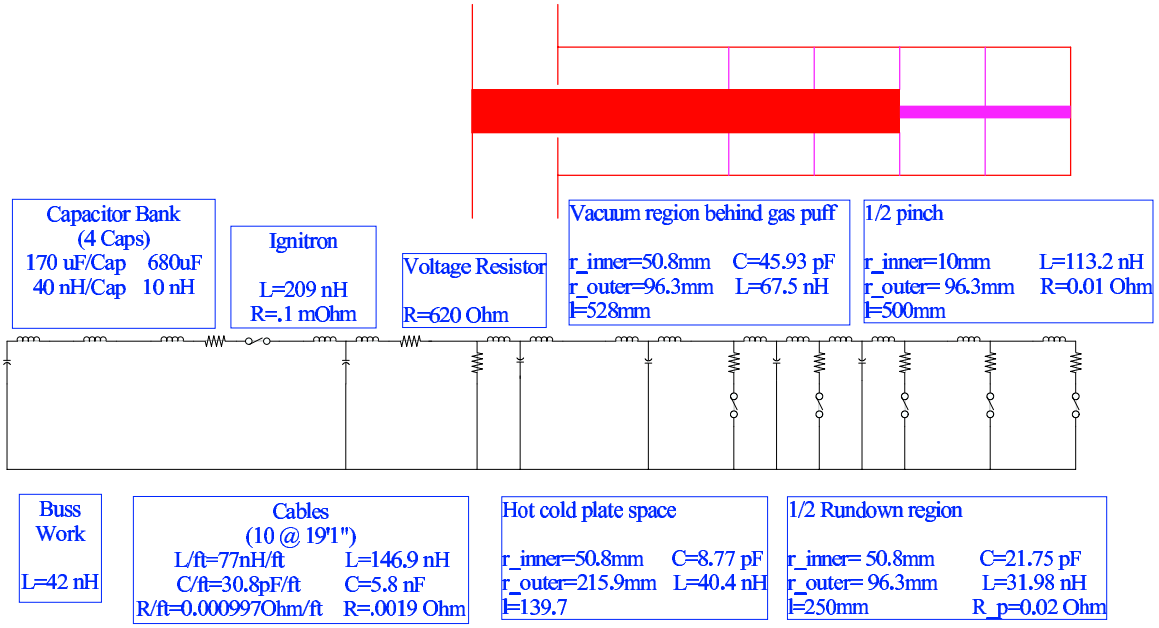

Figure 4.7: Electrical schematic showing the electrical parameters of each component of the original capacitor bank. The dimensions used to calculate the values are also included. The cold plate, hot plate, inner and outer electrodes are shown in red. The magenta lines correspond to the location of plasma current for various switches used to model the total current.

current exceeds 250 kA, the ignitors in the ignitron become wetted. This problem has been solved by limiting the maximum current of each ignitron to less than 200 kA. Forty, 20 feet long RG 217 can be used to connect the ignitron(s) to the hot and cold plates. Twenty cables are presently installed because only one half of the ultimate number of capacitors are being used. The cables are connected symmetrically around the hot and cold plates with brass connectors, to ensure a uniform current.

The original capacitor bank configuration delivered an exponentially decaying, sinusoidal current to the plasma. The circuit parameters of the original configuration are shown in Fig. 4.7. These values were used to simulate the measured currents of calibration and plasma pulses. One method of simulating the varying inductance of the experiment is to trigger the switches shown in Fig. 4.7 at different times. The voltage measured in the experiment cannot be modeled with this method. A better method is to use a variable

Table 4.1: Capacitor bank configurations used in the ZaP experiment. Configurations 4-6 have had the inductance between each capacitor increased to create a pulse forming network (PFN).

Configuration	**Capacitors**	**Ignitrons**	**T/4 μs**
1	4	1	17.6
2	8	1	25.2
3	8	2	20.0
4	4	1 (PFN)	18.2
5	8	1 (PFN)	29.5
6	8	2 (PFN)	20.4

inductor to model the changing inductance from plasma dynamics. This requires the use of a commercial version of SPICE. A pulse forming network was designed using SPICE. Simulations showed that if two ignitrons were triggered at different times a large fraction of the current would not go into the plasma. A better method of forming the flattop on the current is to increase the inductance between the capacitors of each bank. The pulse forming network, PFN, is presently providing a 30-40 μs flattop to the current. The PFN was installed on February 6, 2003.

The six capacitor bank configurations which have been used are shown in Table. 4.1. T/4 is the time for the current to go from 5 to 90 percent of the maximum current. These times are similar when the same number of capacitors and ignitrons are used. The current waveforms are different for the various configurations as shown in Fig. 4.8. The PFN configurations have a lower peak current but longer half cycle times. The current has a flattop with the PFN. Large disruptions of the current are not seen in all of these currents. The variations in the waveforms are due to changes in the gas puff timings and the electrode configuration. The voltage waveforms are shown in Fig. 4.9. The oscillations seen are from plasma dynamics of the Z-pinch. The effect of changing inductance of the experiment is evident in the voltage waveforms. Large voltage spikes, which would indicate instabilities,

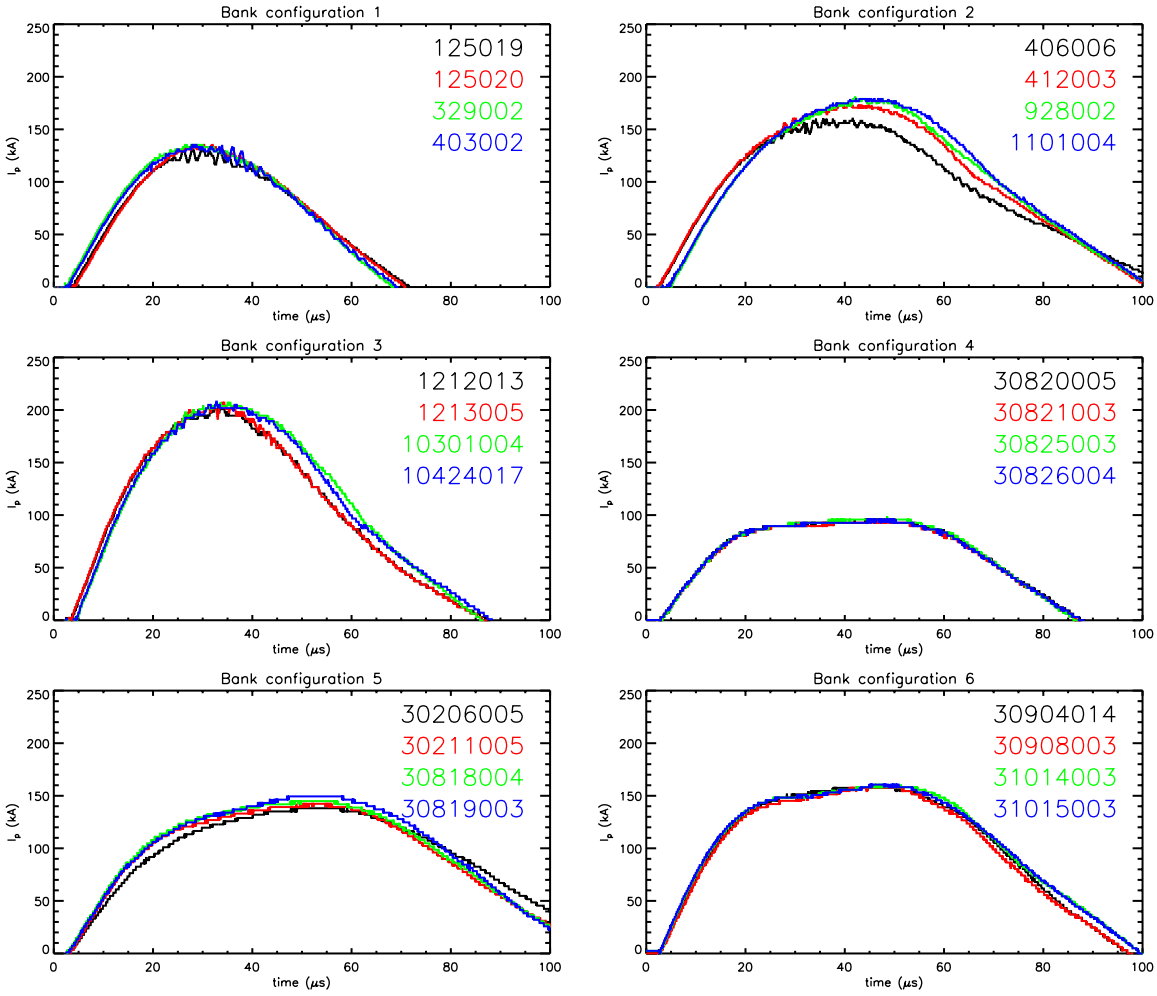

Figure 4.8: The various capacitor bank configurations generate different current waveforms. The capacitor bank was charged to 5 kV for all of these pulses. The first two pulses of each set are from pulses taken shortly after the configuration was installed. The last two are from pulses just before the configuration was changed. In spite of all the waveforms used in the experiment, large disruptions of the current are not seen.

are not seen on these pulses.

The waveforms scale with the charge voltage of the capacitor. The current sheet dynamics change the inductance of the experiment during a pulse. This changing inductance should make the waveforms of the current and the voltage functions of the charge voltage and gas puff timing. The maximum current scales linearly with the charge voltage, shown

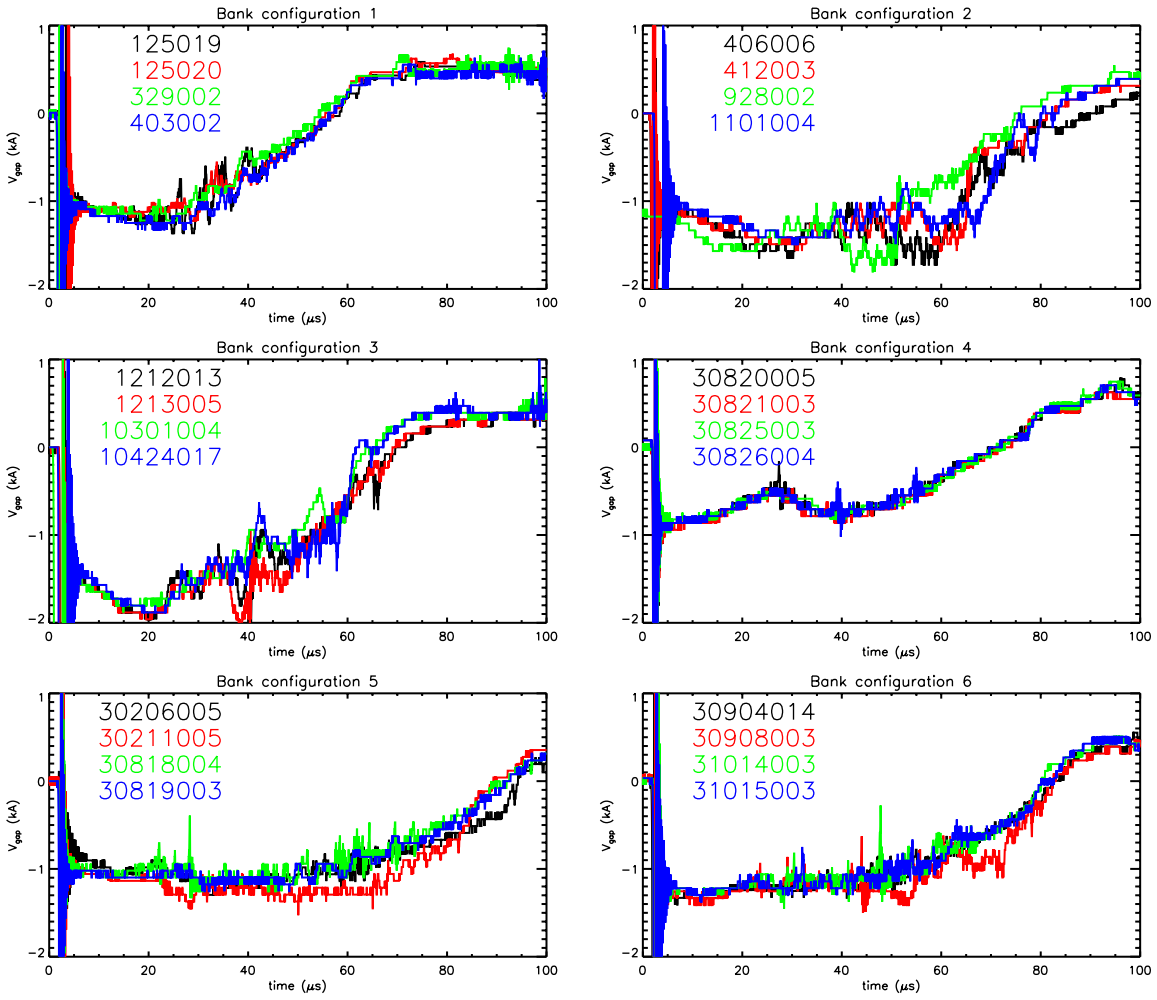

Figure 4.9: The voltage waveforms from the various capacitor bank configurations show the circuit is not an RLC circuit. These are the same pulses used in Fig. 4.8. The large voltage spike at 0 μs is from triggering the ignitron. The change in the waveforms and voltage spikes, seen later in time, are from plasma dynamics.

in Fig. 4.10. The shape of the first quarter cycle of the current also scales with charge voltage. The waveform of the current changes on the higher energy pulses. The outer electrode extension was installed on most of the PFN pulses. The voltage also scales with the charge voltage, shown in Fig. 4.11. The magnitude of the voltage increases on higher energy pulses. This is may from the increasing inductance as the Z-pinch compresses. The scaled bank

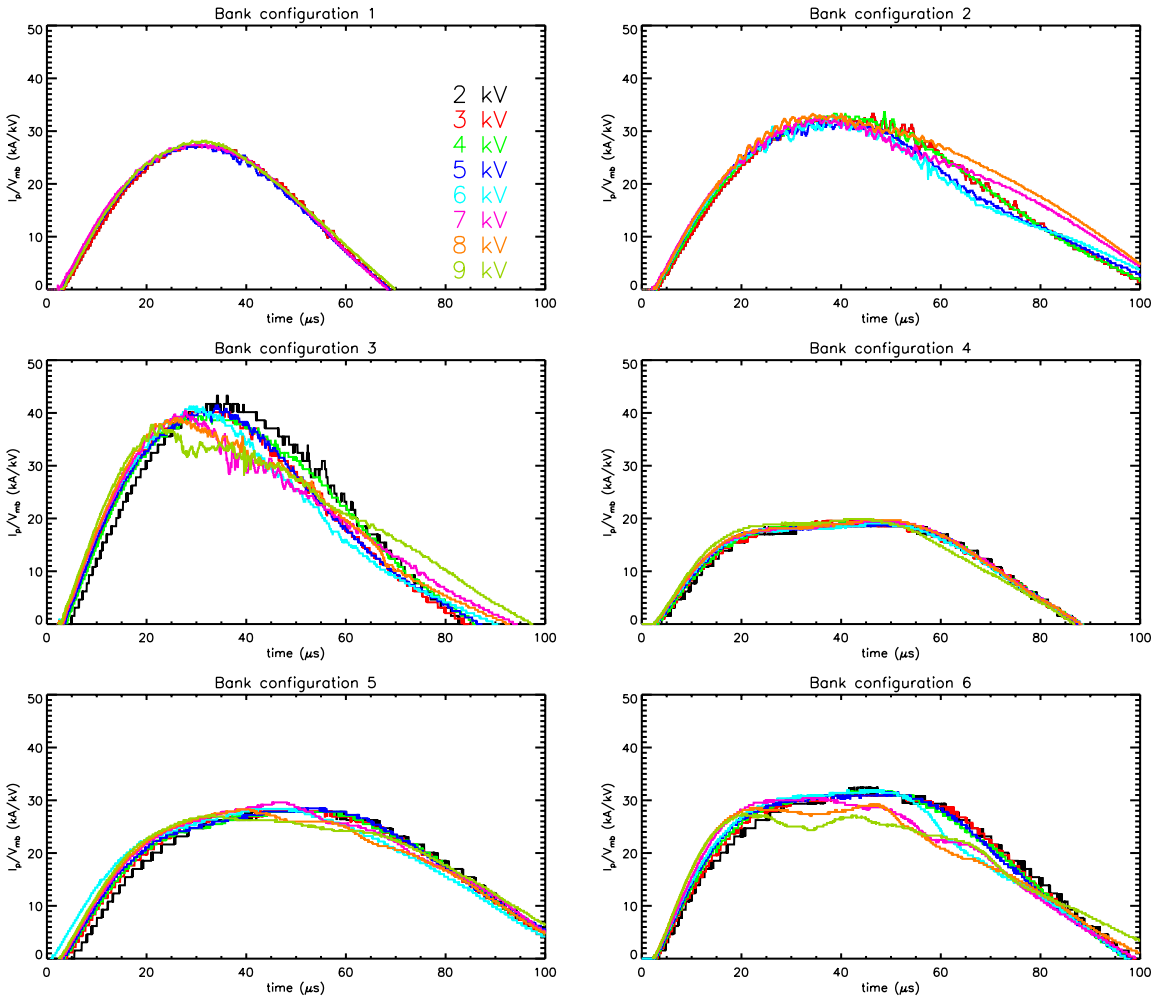

Figure 4.10: The normalized current showing the independence of the current waveform with changing capacitor charge voltage for each bank configuration. The current has been normalized by the charge voltage of the capacitor bank. The charge voltage of each pulse is indicated in the upper left plot. The pulses used of each plot were taken in the same day as the first pulse used in Fig. 4.8. The current wave forms change when the inductance associated with the vacuum region is comparable to the bank inductance.

characteristics show when plasma dynamics begin to have an effect on the circuit.

The original capacitor bank acted as a current source. The impedance of the bank was larger than the impedance of the plasma. The PFN has added a flattop to the current. Future changes to the capacitor bank will have to include the plasma dynamics as the

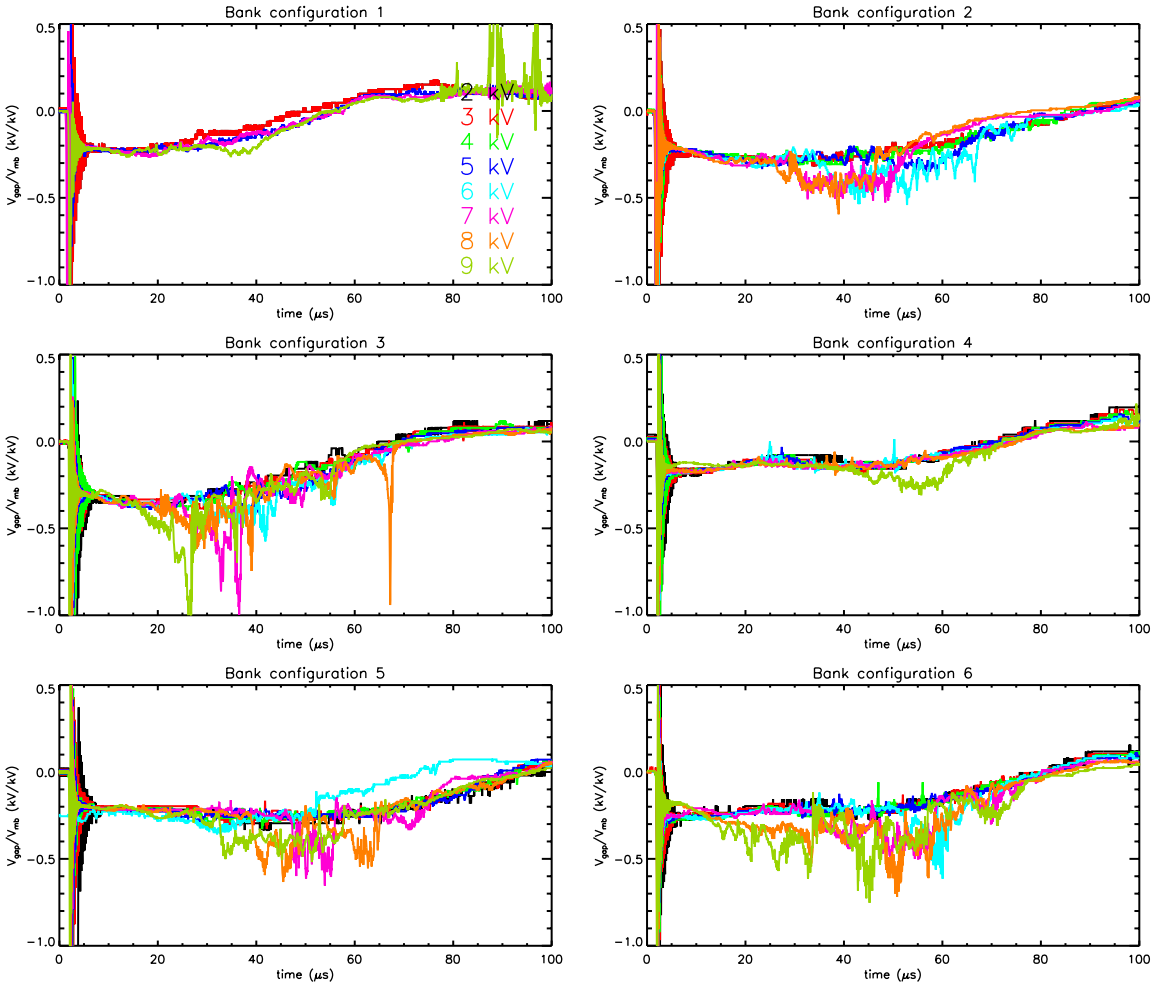

Figure 4.11: The normalized voltage showing the dependence of the voltage waveform with changing capacitor charge voltage for each bank configuration. The voltage has been normalized by the charge voltage of the capacitor bank. The same pulses used in Fig. 4.10 are shown. The voltage waveforms are similar for low bank energies. As the bank energy increases, an increase of the voltage magnitude is seen at 40 μs.

impedance of the plasma and the capacitor bank become comparable.

4.4 Safety Interlocks

A safety interlock system is used to prevent injuries in the lab while the experiment is running. During a pulse life-threatening hazards are present in the lab. The voltages as-

sociated with forming the Z-pinch are lethal. There is an eye hazard from the UV light produced during a pulse and from the high power lasers used in interferometry and holography. Safety stops and door interlocks enable anyone in the lab to shut down the experiment without going near the machine. A personal safety switch is also used to stop the capacitor bank from being charged whenever anyone enters the lab. The grounding rods and battery chargers, which can damage the machine or power in the room, must be put away before all the interlocks can be meet. When all of the interlocks are met the charging supply for the capacitors can be turned on. Whenever an interlock is interrupted the capacitors are discharged through dump resistors. A light on the interlock panel is turned on which informs the operator of the fault. Relays are used on this system since the failure mode can be controlled. On the charge and dump boards the relays are mounted upside down so gravity will always pull the relay into a safe position. The RC time of the dump resistors was chosen so that the capacitors were discharged before anyone could run to the capacitor bank. The hot plate is grounded through the resistor which is used to measure the voltage between the hot and cold plate. A closed circuit television provides the operator with a final visual check for people in the lab before a pulse and allows the operator to look for problems during and after a shot without being in the lab. This system has ensured the safe operation of the experiment.

Chapter 5

DIAGNOSTICS

The properties of the plasma must be measured to gain a physical understanding of the Z-pinch. Measuring of the properties must not affect the plasma. Nonperturbing techniques are used to measure the plasma properties in the ZaP experiment. These techniques measure the properties at the edge of or along a chord through the plasma. Improvements have been made with the reduction and analysis of the data from these techniques.

The data acquisition system is described first. Measurements of the current, voltage, and emission are then described. Section 5.5 describes the surface magnetic field probes. The measured magnetic fields are used to find the location of the Z-pinch and measures the current density in the accelerator. The interferometer, which measures the chord integrated density, is described in Sec. 5.6. Section 5.7 discusses the spectrometers, which are uses to measure the velocity, used on the experiment. A deconvolution technique which calculates the plasma parameters from the measured spectral intensities is derived at the end of the section. Each section of this chapter is divided into the theory , hardware, and analysis of the measurement.

5.1 Data Acquisition

MDSplus is used to set up data acquisition and store the data [61]. The initial pulses, numbered from 1 to 253, were taken with MDSminus. The naming convention for the MDSplus pulses is YYMMDDXXX, where YY is the last two digits of the year, MM is the month, DD is the day and XXX is the pulse number taken on the day. General techniques used to reduce the noise on the diagnostics are discussed. Error analysis and noise calculations common to most diagnostics are also discussed in this section.

5.1.1 Hardware

ZaP uses two main data acquisition systems. Most of the data are taken with 20 MHz digitizers in several CAMAC crates. A PC is used to take the spectroscopy data. ZaP is the first experiment at the University of Washington to exclusively use MDSplus for its data acquisition. MDSplus provides a set of tools for performing data acquisition and analysis for pulsed experiments. The setup values, raw data, scale factors and analyzed data for each pulse is stored in a tree structure (similar to "folders". The tree is made up of nodes which act like a file system. There are two types of nodes, children or branch nodes which are used to define the structure of the tree and member nodes which contain the data. Member nodes can also have members or children below them. Large amounts of data are easily organized and accessed within this structure. The trees can be accessed and modified with a GUI, "TRAVERSER", or a command language, TCL.

There are three phases to a ZaP pulse; Init, Store, Analyze. All of these phases are controlled by nodes in the tree. During the Init phase the digitizers are set up. The data are read and written to the appropriate nodes during the Store phase. During the Analysis phase, data reduction programs are run. Data from spectroscopy are added after it is reduced with an IDL program.

All of the data in the tree can be accessed with the same routines. While pulses are being taken, the data are checked with a plotting routine, DWSCOPE, which can monitor up to 64 traces. Normally a number of scopes are used to view different aspects of the plasma. IDL routines are also used to access and analyze the data.

5.1.2 Noise Reduction

ZaP is a coaxial machine with ideally no stray magnetic fields. Even though electromagnetic noise should not be a problem on ZaP, ground loops are still avoided on the experiment. The screen room ground is used as the reference ground potential. All grounds in the lab are connected once and only once to the screen room. An easy test for ground loops is to remove the ground and check for continuity with the screen room. If the continuity is not broken or as a check of the hardware grounds, a ground loop chase is performed. An audio

frequency oscillating magnetic field is driven around a ground strap. If there is a ground loop, the changing flux through the loop generates a current along the ground loop. The generated current in the loop can be picked up with a small coil amplified by a transistor radio. Since the entire loop can be mapped with this process, the ground loop is broken at an appropriate point.

Most of the noise seen on ZaP is electrostatic. Enclosing the hot plate in a Faraday shield has helped to reduce the electrostatic noise. However better shielding is needed on the interferometer system and the PMT's. The coax cables for these devices are placed inside a copper braid which is attached to the screen room. The electronics in the lab are then placed in a conducting box attached to the braid. This effectively brings the screen room out to the electronics which reduces the noise. Since the coaxial cable is enclosed by the shield ground loops between the cables within the braid are not a problem.

Another method of reducing the electrostatic pick up is to use common mode filters on the coaxial cables at the top of the screen room. The cables are wrapped 10 times around five ferrite cores. Currents due to common mode electrostatic pick-up, which travel in the same direction on the center conductor and the shield, are attenuated by the inductance of the winding. Signal currents are not affected by the winding because they are in opposite directions around the winding creating no net flux.

A brute force method of reducing the noise is to increase the signal level. If possible, the signal levels generated in the lab are made larger than the digitizers can measure. The noise pick up level is constant. Attenuators are used in the screen to decrease the signal level to the maximum voltage of the digitizer. The noise level is effectively diminished to an acceptable level. This last resort method has worked well with various diagnostics on ZaP.

5.1.3 Error and Noise Analysis

Every measurement has an associated uncertainty. These errors must be propagated through the analysis of the data to find the error bars. Methods of finding the error bars are described in Bevington [3]. The error, σ_{ave} of an average is

$$\sigma_{\text{ave}}^2 = \frac{1}{N-1}\sum(x-\bar{x})^2 \tag{5.1}$$

where N is number of measurements, and $\bar{x}$ is the average. When a function is given by $y = f(u, v)$, the error, σ_y, is

$$\sigma_y^2 = \sigma_u^2 \left(\frac{\partial y}{\partial u}\right)^2 + \sigma_v^2 \left(\frac{\partial y}{\partial v}\right)^2 + \sigma_{uv}^2 \left(\frac{\partial y}{\partial u}\right)\left(\frac{\partial y}{\partial v}\right) \tag{5.2}$$

where σ_u and σ_v are the uncertainties in u and v and $\sigma_{uv} = 0$ when u and v are not correlated. The terms used in these equations are discussed in the appropriate sections.

The digitizers record both the signal and noise in the signal. The electrostatic noise from the experiment has a high frequency and is time dependent. All of the digitized information is used in the analysis of the data when digital filters are used to calculate the signal and the noise. A low pass filter is used to find the signal. The frequence of the filter, f_{filter}, is diagnostic dependent. A high pass filter is then used to find the noise. The time variation of the noise is found a $1/f_{\text{filter}}$ window of the amplitude of the high frequency data and Eq. 5.1. This method of finding the signal and uncertainty in the signal uses of all the recorded data.

5.2 Rogowski Coils

A standard method measuring the current is with a Rogowski coil [27]. The coil measures the flux from a changing current. The coils are made by wrapping a wire around a section of RG58 cable with the shield removed. They are calibrated by comparing the current measured with the coil to the current measured with a calibrated current transformer.

5.2.1 Theory

The Rogowski coil uses Ampere's law

$$\oint \boldsymbol{B} \cdot d\boldsymbol{l} = \mu_0 I \tag{5.3}$$

to measure the enclosed current, I. The coil consists of a form with a number of windings per unit length, n, which is completely surrounds the current shown in Fig. 5.1. The coil should either be back-wound or have the return wire come back through the coil to avoid measuring changing fluxes through the enclosed area. The area of the coil and spacing

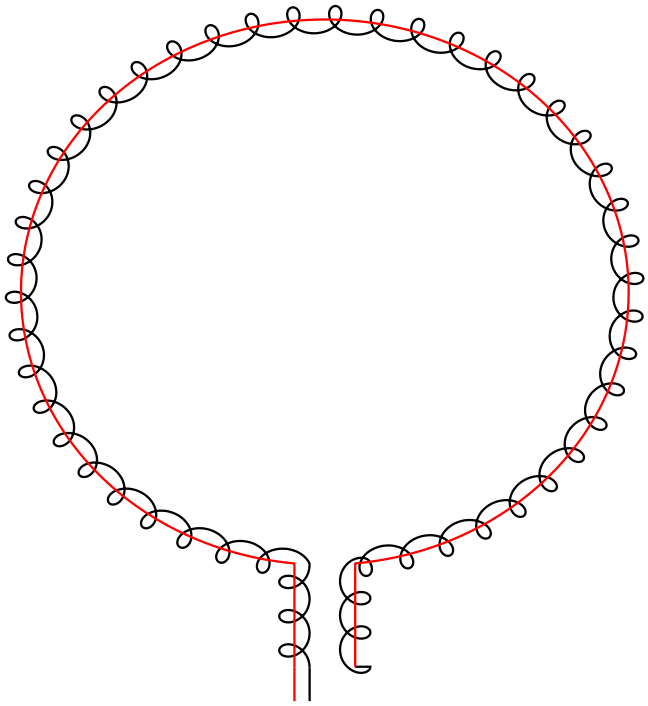

Figure 5.1: The Rogowski coils have the return through the coil. The windings are shown in black. The return wire, shown in red, is run back through the coil preventing a changing flux through the enclosed area from affecting the measurement. It is also important that both ends of the coil meet as shown.

between the turns are kept small to avoid large gradients in the magnetic field. A changing current through the loop causes a changing flux through each turn of the coil. The changing flux causes a voltage from the coil given by

$$V = nA\mu_0 \dot{I} \tag{5.4}$$

where n is the number of turns per unit length and A is the area of the winding. By integrating the voltage, the digitized signal is proportional to the enclosed current.

The voltage can be integrated actively, passively or a combination of the two [49, 50]. Rogowski coils have an inductance and capacitance associated with them. The inductance is given by

$$L = l\mu n^2 A \tag{5.5}$$

where l is the length of the coil. They can be made self-integrating with the proper termination. ZaP uses active integrators, DAFI's, to integrate its signals. The Rogowski coils are designed to remove any self integration, by adding a series resistor until the waveform from the current transformer and the integrated Rogowski signal differ only by a scale factor.

5.2.2 Hardware

An easy method of making a Rogowski is to use coaxial cable as the form. The outer shield of a RG58 cable of the desired length, 61 cm, of the coil is removed. The Rogowski coils are made by wrapping 32 gauge wire around the exposed insulator. This gives an area of 6.7 mm^2, $n \approx 4.3\ \text{mm}^{-1}$ and $L = 95\ \mu\text{H}$. The ends of the wire are soldered to the shield and inner conductor of the RG58 cable which is inside the coils. Clear heat shrink is then used hold the windings in place. The Rogowski is wrapped around the desired current as shown in Fig. 5.1. The voltages induced from the fluxes through the straight section cancel each other when the ends of the coil stop at the same location. The voltage is integrated in time using a DAFI and digitized. The digitized signal is proportional to the enclosed current. Rogowski coils are placed around the inner electrode, each ignitron, and outside the hole in the endwall to measure the various total currents in the experiment.

5.2.3 Calibration

The Rogowski coils are calibrated using current transformers. The currents generated by the ZaP experiment are larger than a current transformer can measure. Ten coaxial cables are used to supply the current to the hot and cold plates. The current through one of the cables can be measured by a current transformer. Three current transformers spaced at equal angles around the hot plate are used to measure the current and check the symmetry of the current at the hot plate. The three current transformers each measured the same current indicating that the current is equally divided between the ten cables. The total current through the inner electrode is then 10 times the current measured by each current transformer. The calibration factor between the digitized voltage and enclosed current is then found.

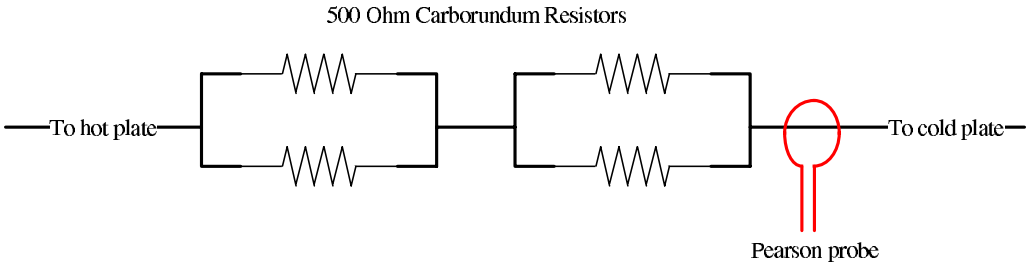

Figure 5.2: The voltage between the hot and cold plates is measured with a known resistance and a current transformer. The total resistance of the system is 500 Ω. This resistance is much larger than the total plasma resistance. The current through the resistors is monitored removing the need to made connections between the hot or cold plates and the screen room.

The Rogowski coils self-integrate when terminated with 50 Ω, $\tau_{L/R} = 1.9$ μs. A phase shift is seen between the current transformer signal and the Rogowski signal. Various values of series resistance were tried. The phase shift and signal decrease with increasing resistance. A series resistance of 1 kΩ removes the phase shift and increases the frequency response without decreasing the signal to a low level.

5.3 Voltage Measurement

The voltage is measured by measuring the current through a known resistance across the hot and cold plates. This method has the advantage that there are no connections between the hot or cold plates and the screen room. Calibration of the system is done by charging the hot and cold plates without making a plasma.

5.3.1 Hardware

The voltage measurement consists of a set of resistors and a current transformer. Four 500 Ω resistors are used. Two sets of two resistors in parallel are used in series giving a total resistance of 500 Ω shown in Fig. 5.2. These are placed across the hot and cold plates. A current transformer is then used to measure the current though the resistors. The current through the resistors is small compared to the plasma current. The voltage from the current

transformer is then digitized.

5.3.2 Calibration

The voltage measurement is calibrated by applying a known voltage between the hot and cold plate. Ideally 10 kV can be applied between hot and cold plate without arcing. It was decided to apply the voltage for a short time and then form a plasma to avoid the damage which an arc would cause. The capacitor bank was charged to 2 kV. The gas puff timing was adjusted such that there was a 50 μs delay between the triggering of the ignitron and formation of a plasma. Initially, ringing is seen in the voltage measurement due to the mismatch of the impedances of the cables and electrodes. The signal reaches a steady state which corresponds the a voltage of 2 kV. The calibration factor is found with the data from this time. A plasma is then formed protecting the experiment from an uncontrolled arc.

The scale factor from the empirical method agrees with a calculated scale factor. The measured resistance of the system in 620 Ω. The scale factor of the current transformer is 0.05 V/A. A 10 dB attenuator is used to reduce the voltage at the digitizer. The calculated scale factor, 39,129 V/V, is within a percent of the measured factor, 39,216 V/V.

5.4 Imacon Camera and Emission Monitors

Light from the plasma can provide information about the source of origination. There are two main sources of light from ZaP, line radiation and bremsstrahlung. This section discusses using the radiation to describe the shape and general characteristics of the plasma. A detailed line radiation discussion is found in Sec. 5.7. These diagnostics are used to confirm measurements made with other diagnostics.

5.4.1 Theory

Most of the light emitted from a plasma is from line radiation due to impurities. Electrons in partially stripped ions or neutrals are excited by collisions with free electrons. When they fall to a lower energy level photons are emitted with a wavelength, λ_0, given by

$$\lambda_0 = \frac{hc}{E_k - E_i} \tag{5.6}$$

where $E_k,\ E_i$ are the initial and final energy levels. An interesting line to monitor is the H_α line at 656.28516 nm from neutral hydrogen. This line is emitted when neutral hydrogen penetrates the plasma. The intensity of this line is the largest when the plasma is cold. The heating and cooling of the plasma can be inferred from the time history of this line. Pictures of the H_α line show the edge of the pinch.

Another source of radiation is bremsstrahlung radiation. This is from the "braking" of the electrons when they collide with ions. The spectral power emitted into 4π sr per unit frequency is given by [27]

$$4\pi j(\omega) = n_e n_i Z^2 \left(\frac{e^2}{4\pi\epsilon_0}\right)^3 \frac{16\pi}{3\sqrt{3}m^2c^3}\left(\frac{2m}{\pi T}\right)^{1/2} e^{-\hbar\omega/T}\bar{g}\ \left[\frac{W}{\frac{rad}{s}\ m^3}\right] \tag{5.7}$$

where all variables are in SI units, T is in joules and the Maxwell-averaged Gaunt factor, $\bar{g}$, can be approximated by

$$\bar{g} = \begin{cases} \frac{\sqrt{3}}{\pi}\ln\left|\left(\frac{2T}{\zeta m}\right)^{3/2}\frac{2m}{\zeta\omega}\left(\frac{4\pi\epsilon_0}{Ze^2}\right)\right| & : T \ll Z^2 Ry \text{ (Kramer)} \\ \frac{\sqrt{3}}{\pi}K_0\left(\frac{\bar{\omega}}{2T}\right)\exp\left(\frac{\bar{\omega}}{2T}\right) & : T \gg Z^2 Ry \text{ (Born)} \\ \frac{\sqrt{3}}{\pi}\ln\left|\frac{4T}{\zeta\hbar\omega}\right| & : T \ll Z^2 Ry \text{ (Born)} \end{cases} \tag{5.8}$$

where K_0 is the modified Hankel function and $\zeta = 1.78$ is the reciprocal of Euler's constant [27]. The bremsstrahlung radiation is large when viewing a dense plasma. Photos of the bremsstrahlung radiation show the shape of the emission from the Z-pinch.

5.4.2 Hardware

Filter scopes are used to monitor the radiation from a narrow wavelength region. A collimator is used view plasma through one of the optic ports shown in Fig. 5.3. The collimator consists of a piece of 1 inch thick Delrin®, black plastic, with two 3.175 mm diameter holes through it. Band pass filters are placed in front each hole to measure the H_α and bremsstrahlung radiation. The H_α filter is centered at 656.5 nm with a 9.5 nm half band width, HBW (Coherent Auburn Group, part number 35-3995). The bremsstrahlung filter is centered at 630.2 nm with a 11.1 nm HBW (Coherent Auburn Group, part number 35-3896). No line radiation is measured with the spectrometers in this region. Photodetectors

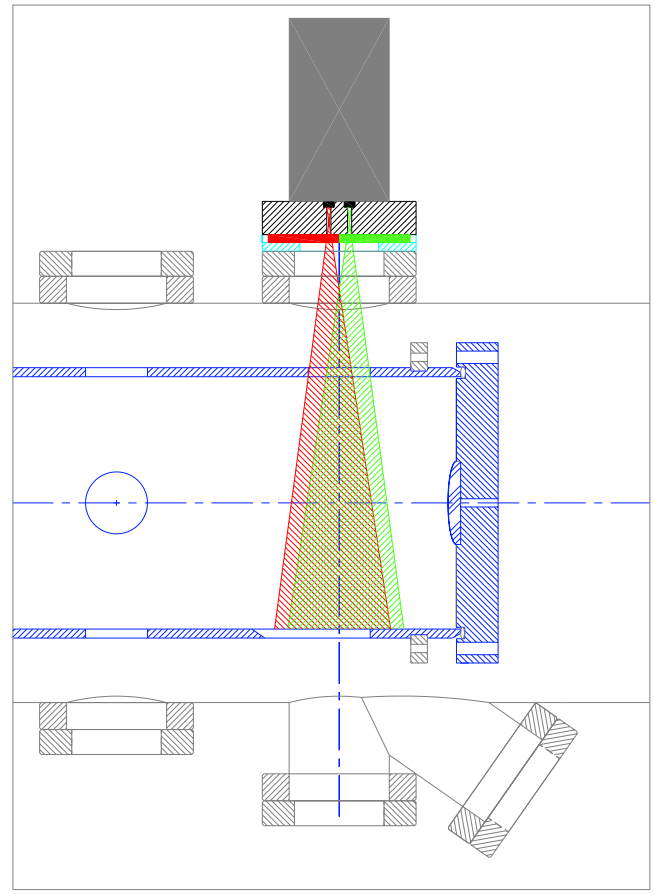

Figure 5.3: The ray trace from the filterscopes is shown in a cutaway section of the experiment (grey). The collimator (black) limits the view of the detectors (solid black) such that the exposed copper of the outer electrode is not seen. Band pass filters, H_α (red) and Bremsstrahlung (green) are used limit the wavelength region measured by each detector. Both filterscopes view approximately the same region of plasma.

are used at the opposite side of the hole to measure the intensity of the light. These photodetectors and electronics are identical to the detectors used on the interferometer, shown in Fig. 5.4. The signal from each of these is then digitized. These measurements show the arrival, heating and disruption of the Z-pinch.

The Imacon model 790 camera takes images of the plasma. The images are electronically moved to different regions of a piece of film. The time between images, 500 ns, 1 μs, or 2 μs is specified by different modules. Photos are taken through various windows on the

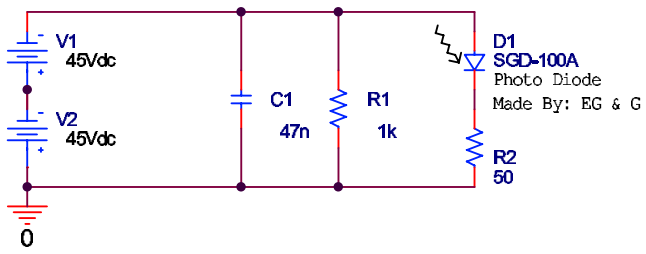

Figure 5.4: Shown is the electrical schematic of the photo detectors used on the ZaP experiment. The detectors (EG&G SGD-100A) are biased with -90V to increase their linearity and frequency response. Capacitors are used to reduce the noise of the detector and provide a fast source of high current. The voltage is measured across R2.

vacuum tank. Different filters are used to view various regions of the plasma. An H_α, 610 nm high pass (wavelength) , and a "bremsstrahlung", 500 nm to 600 nm, filter are used on the camera. While these filters are not ideal, the main line radiation measured by the H_α filter is H_α and the radiation measured from the 'bremsstrahlung' filter is not H_α. The images are record on 3200 speed Polaroid 667 film. The photos provide information on the initial breakdown of the gas and shape of the Z-pinch light emission throughout its various phases.

5.5 Magnetic Surface Probes

Knowledge of the magnetic field provides information about the current sheet and the Z-pinch. Surface probes along the outer electrode provide a means to measure the magnetic field structure along the experiment on every plasma pulse. By keeping the design of these probes simple, they are easily massed produced. The probes are calibrated *insitu* to minimize wall effects. The calculation the current sheet characteristics, mode magnitude, current density and Z-pinch location are discussed at the end of this section.

5.5.1 Theory

A varying magnetic field can be measured with a coil of wire. The changing magnetic is creates a voltage in the coil according to Faraday's Law.[27]

$$\oint_c \mathbf{E} \cdot d\mathbf{l} = -\int_s \dot{\mathbf{B}} \cdot d\mathbf{s} \tag{5.9}$$

By integrating the voltage with an analog integrator, the digitized voltage is

$$V_0 = \frac{NAB}{RC} \tag{5.10}$$

where V_0 is the measured voltage, N is the number of turns of the probe, A is the area of the probe, B is the average field over the area of the probe, and R, C are the resistance and capacitance of the integrator. Surface mounted magnetic probes measure the magnetic field which penetrates holes in the material walls. The attenuation of the magnetic field from the walls was accounted for in the calibration of the probes. No corrections have been made for the frequency response of the probes.

5.5.2 Hardware

One array of single winding (θ) and four arrays of double winding (θ, z) surface mounted probes are used to measure B_θ and B_z on the ZaP experiment. These arrays are located on the outer electrode. A linear array of single winding probes is located 22.5 degrees below the horizontal plane on the $-x$ side of the experiment. The probes are placed 5 cm apart from $z = -80$ cm to $z = 20$ cm. Two additional probes are located at $z = -120$ cm and $z = -100$ cm. The array is continued on the 50 cm extension with probes located 5 cm apart from $z = 35$ cm to $z = 70$ cm. This array is used to measure the axial location and distribution of the current sheet. Two azimuthal arrays of double winding probes are also used on the original electrode. One of these azimuthal arrays is located at the center of the pinch, at $z = 0$ cm. This array has eight probes located 45 degrees apart, and is used to measure the mode activity at the center on the pinch. The other azimuthal array is located at the exit of the acceleration region at $z = -25$ cm. This array is similar to the one located at the center of the pinch except the probes at the top of the electrode have been removed to

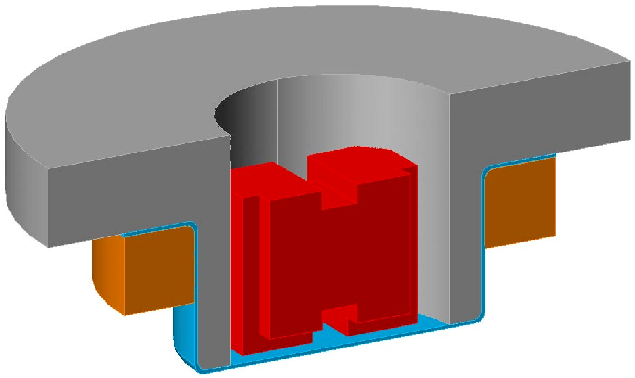

Figure 5.5: Shown is a cut away of two winding probe. The stainless steel hat is grey, the copper ring is orange, the Kel-F form is red and the Ta foil is blue. The windings are wrapped around the groves in the Kel-F form (not shown). The single winding form is identical to the two winding form with the inner winding missing.

allow for interferometry access. The array at the muzzle is used to measure the symmetry of the current sheet, as well as the location of the inner electrode attachment point. Two additional azimuthal arrays, identical to the $z = 0$ cm azimuthal array, are located on the 50 cm extension at $z = 35$ cm and $z = 70$ cm.

All surface probe assemblies are identical to the one shown in Fig. 5.5. The probes have 10 turns of 32 gauge wire around a Kel-F® form. The form is placed in a stainless steel hat. Tantalum foil is stretched across the cup and held in place with a copper washer, protecting the assembly from the plasma. The assembly is held into holes made in the outer electrode with two silver plated through-holed bolts. The leads from the probes are twisted together and connected to DB25 vacuum feedthroughs. The leads are held off the electrode with stainless steel 'telephone poles' and the exposed leads are shielded from the plasma with stainless steel boxes and stainless steel foil. The signals from the probes are integrated with analog integrators (DAFI's). The integrated signal is then digitized.

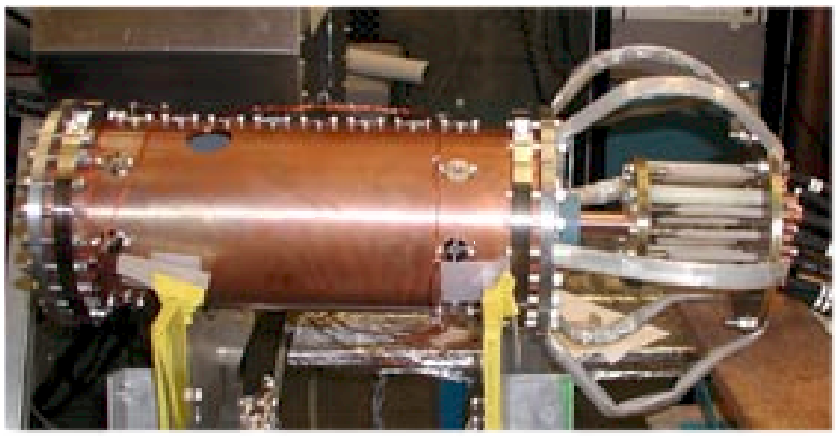

Figure 5.6: Shown is a picture of the electrode extension with the calibration rod attached. The linear array is on top of the electrode. The two azimuthal arrays can be seen near the stainless steel split rings. Aluminum plates are used as end walls. The calibration rod can be moved off the axis of the electrode by adjusting the blue nylon fitting. The header has been designed to ensure that the currents entering the electrode are symmetric.

5.5.3 Calibration of the Probes

The single-winding probes were aligned and calibrated with an axial current at the center of the outer electrode. A copper rod was used to simulate plasma. One end of the rod was attached to the outer electrode as shown in Fig 5.6. An attempt was made to align the probes by driving oscillating current through the rod. The probes could rotated by a few degrees and measure the same flux, since the flux is proportional to the cosine of the angle between the windings and the z axis. Aligning the probes visually, by lining up the wires on the probe with a straight edge, parallel to the z axis, was found to be the most consistent method. The calibration factors for converting the measured flux to magnetic field are then found. The probes are connected to the data acquisition system. A known current is driven through the rod. The scale factor between the measured flux and the magnetic field the current would produce at the surface of the outer electrode is found for each probe. An IDL routine, *cal_axial_b_cal_fact_11214.pro* Sec. 220, which finds the peak of the magnetic field, and the peak of measured flux is used to find the scale factors for probe.

The two-winding probes are calibrated in a similar fashion [5]. Two sets of calibration

data are needed to find the transformation matrix, $\mathbf{A}$

$$\begin{bmatrix} B_\theta \\ B_z \end{bmatrix} = \begin{bmatrix} A_{11} A_{12} \\ A_{21} A_{22} \end{bmatrix} \begin{bmatrix} \Phi_\theta \\ \Phi_z \end{bmatrix} \tag{5.11}$$

where $\mathbf{B}$ is the magnetic field and $\mathbf{\Phi}$ is the measured flux. The inverse of $\mathbf{A}$ can measured by applying a known magnetic field and measuring the fluxes. The magnetic fields are oriented such that one of the fields is zero. One column of $\mathbf{A}^{-1}$ is then found. A perpendicular of the magnetic field is then used to find the other column. Initially the theta windings are oriented perpendicular to the minus z axis. This is done with a straight edge perpendicular to the z axis. A known current is driven through the rod and the fluxes of both windings were recorded. The theta windings are then oriented parallel to the z axis. A known current is again driven through the rod and the fluxes of both windings are recorded. The transformation matrix is then found.

Since the probes are located inside the outer electrode wall, the scale factors have a frequency dependence. High frequency magnetic fields are attenuated by the electrode. Presently in the experiment no corrections have been made for the frequency response of the probes. A single scale factor, corresponding to the current frequency, is used. Simulations are being performed in ANSYS and FEMLAB to model the frequency response. Future calibrations which will measure the frequency response of the probes are being designed.

5.5.4 Reduction of Azimuthal Array Data

The azimuthal arrays can give the location of the plasma current. Interpretation of the structures is simplified by looking at the magnitude of the Fourier components of the magnetic field. The normalized components can be related to the position of the plasma for a constant displacement of the current along the z axis. When the displacement of the current has a z dependence, the normalized mode amplitudes decrease.

Calculation of the Fourier Components

The Fourier components of the magnetic field are found using a least squares fit. The magnetic data are assumed to be of the form

$$B_\theta(\theta_i) = \sum_{j=0,m} a_j \cos(j\theta_i) + \sum_{j=1,m} b_j \sin(j\theta_i) \tag{5.12}$$

where θ_i is the azimuthal location, in radians, of the probe. When a discrete number, n, of measurements are made, equation 5.12 can be written as

$$\mathbf{B} = \mathbf{AC} \tag{5.13}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & \cos\theta_0 & \dots & \cos m\theta_0 & \sin\theta_0 & \dots & \sin m\theta_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos\theta_i & \dots & \cos m\theta_i & \sin\theta_i & \dots & \sin m\theta_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos\theta_n & \dots & \cos m\theta_n & \sin\theta_n & \dots & \sin m\theta_n \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} B_\theta(\theta_0) \\ \vdots \\ B_\theta(\theta_i) \\ \vdots \\ B_\theta(\theta_n) \end{bmatrix} \quad \text{and,} \quad \mathbf{C} = \begin{bmatrix} a_0 \\ \vdots \\ a_i \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{bmatrix}.$$

The Fourier components at each time are then found with [64]

$$\mathbf{C} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{B} \tag{5.14}$$

where $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ is the pseudoinverse, $\mathbf{A}^*$, of $\mathbf{A}$. $2m+1$ measurements are required to resolve up to mode m. When the probes are evenly spaced around a circle ($\theta_i = i2\pi/n$), $\mathbf{A}^T\mathbf{A}$ is diagonal. Since $\mathbf{A}$ is only dependent on the geometry of the probes, $\left(\mathbf{A}^T\mathbf{A}\right)^{-1}$ is calculated once with an IDL code *find_mode_array.pro* Sec. C.2.3. The amplitude, m_i, and the phase, ϕ_i, of mode i are given by

$$\begin{aligned} m_i &= \sqrt{a_i^2 + b_i^2} \\ \phi_i &= \arctan(b_i, a_i) \end{aligned} \tag{5.15}$$

The error in the components are given by

$$\triangle \mathbf{C}_j^2 = \sum_{i=0,n} \left(\mathbf{A}_{i,j}^* \triangle \mathbf{B}_i\right)^2 \tag{5.16}$$

An IDL code, *mode_calculation.pro* Sec. C.2.4, is used to calculate the Fourier components and put them into the MDSplus tree.

Calculation of the Position of the Plasma

The position of the current is related to the normalized m_1 data. The normalized is defined as

$$\text{normalized } m_i \equiv \min\left(\frac{m_i}{\max\left(m_0, 0.01\right)}, 1.0\right). \tag{5.17}$$

In a flux conserver, the current is modeled as a current along the z-axis which attaches to the electrodes and a loop current inside the flux conserver which follows the displacements and returns along the z axis. The magnetic field at the wall, $\mathbf{B}$, is given by (see Appendix A)

$$\mathbf{B}_\perp^{(1)} = 0 \tag{5.18}$$

$$\mathbf{B}_\parallel = \mathbf{B}_\parallel^{(0)} + 2\mathbf{B}_\parallel^{(1)} \tag{5.19}$$

where $\mathbf{B}^{(0)}$ and $\mathbf{B}^{(1)}$ are the magnetic field from the axial current and loop of current respectively. For an infinitely long pinch displaced $\triangle r$ in the $\theta = 0$ direction the magnetic field at the wall is given by

$$B_\theta = \frac{\mu_0 I}{2\pi}\left(\frac{1}{r_{\text{wall}}} + 2\frac{r_{\text{wall}} - \triangle r\cos\theta}{r_{\text{wall}}^2 - 2r_{\text{wall}}\triangle r\cos\theta + \triangle r^2}\frac{1}{r_{\text{wall}}}\right) \tag{5.20}$$

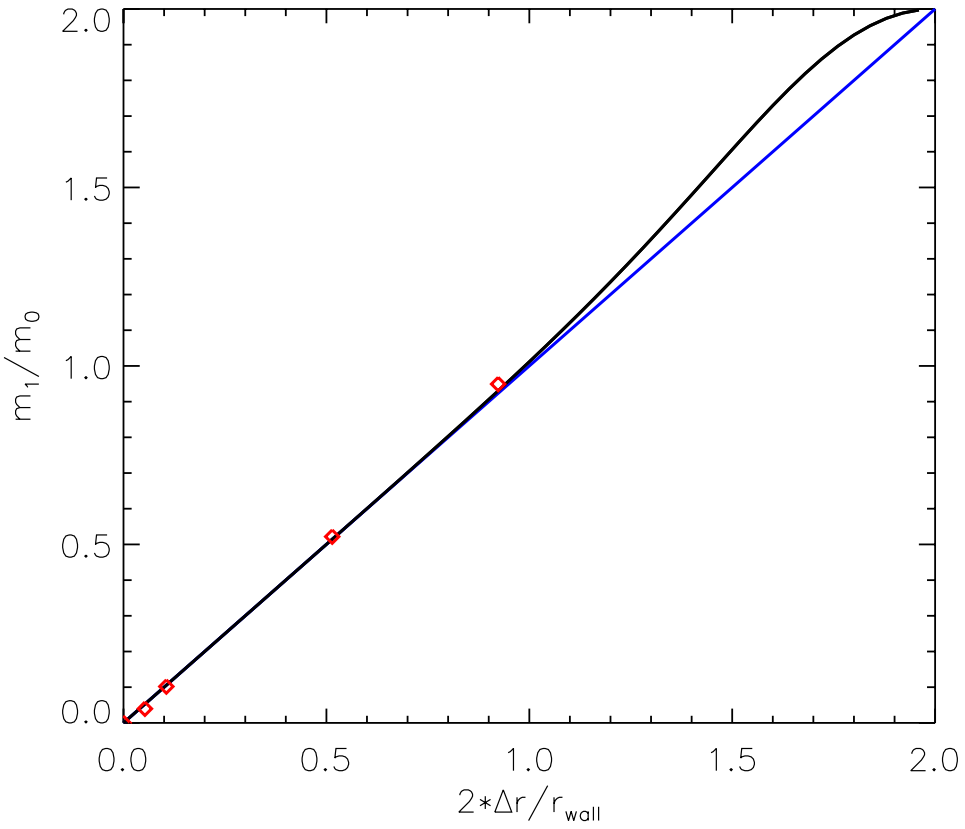

Figure 5.7: The normalized m_1 is proportional to a constant axial displacement of the Z-pinch. The black line is the normalized m_1. The blue line is from Eq. 5.22. The red diamonds are experimental values from moving the calibration rod off axis measuring m_1. The theoretical and measured values agree.

Which can be written as

$$B_\theta = \frac{\mu_0 I}{2\pi}\frac{1}{r_{\text{wall}}}\left(1 + 2\frac{\triangle r}{r_{\text{wall}}}\cos\theta\right) \tag{5.21}$$

when $\triangle r/r_{\text{wall}} \ll 1$. The normalized m_1 is related to the displacement of the current by

$$\frac{m_1}{m_0} = 2\frac{\triangle r}{r_{\text{wall}}} \tag{5.22}$$

Detailed calculations show that this relation is valid for $\triangle r/r_{\text{wall}} \sim 0.5$ as shown in Fig. 5.7.

When the displacements have an axial structure this relationship sets the lower bound on the maximum displaced current. Shown in Fig. 5.8 are contours of normalized m_1 for

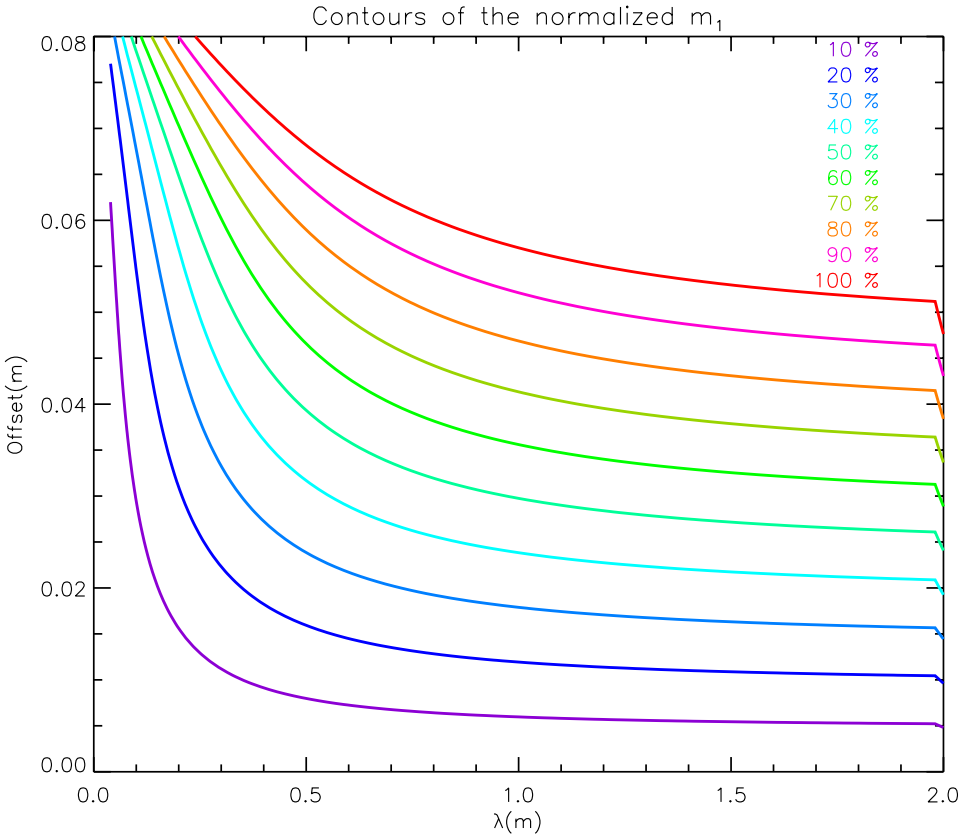

Figure 5.8: The normalized m_1 is a function of the wavelength, k, and the amplitude when the perturbation is one the half of a cycle. The perturbation moved towards the $\theta = 0$ probe and is centered about the probe array. Notice that the normalized mode values decrease for short wavelength perturbation. The upper limit of the normalized m_1, for a given displacement, is given by Eq. 5.22.

a single half wavelength perturbation of the current. The azimuthal array was placed at the maximum displacement location. The magnetic field at each probe was found using Eq. 5.18 for various k's and maximum displacements of the perturbation. The amplitudes of the modes are then found using Eq. 5.14. The contour plot shows that Eq. 5.22 gives a lower bound on the displacement of the current. The measurement of the wavelength of the modes is work being presently pursued with internal magnetic probes and a linear array of photodiodes.

5.5.5 Reduction of Axial Array Data

The axial array measures the z dependence of the azimuthal magnetic field. The axial location of the current sheet of is found by determining when the magnetic field begins to increase along the electrode. The velocity, acceleration, and mass of the current sheet can then be found. With a few assumptions, which are valid in the acceleration region, the radial current density at the outer electrode is measured at the outer electrode.

Current sheet velocity measurements

The location of the current sheet along the z axis is measured by finding the time the magnetic field begins to increase. Once the location of the current sheet is known the velocity and acceleration of the current can be calculated. The total mass of the current sheet can also be estimated if the current sheet is assumed to be a thin washer of current pushed by a magnetic piston. The washer is pushed by the magnetic pressure behind it and accumulates mass as it ionizes the gas ahead of it. With the flat washer approximation, the total impulse, J_{imp}, of the current sheet is given by

$$J_{imp} = \int_0^t \frac{\pi r_{\text{wall}}^2 \left(B_\theta^{(-120)}\right)^2}{\mu_0} \ln \frac{r_{\text{wall}}}{r_{\text{inner}}} dt \tag{5.23}$$

where $B_\theta^{(-120)}$ is the magnetic field measured at $z = -120$ cm, r_{wall} is the radius of the inner electrode and r_{inner} is the radius of the inner electrode. The mass of the washer, m_{cs}, is then given by

$$m_{cs} = \frac{J_{imp}}{v_{cs}} \tag{5.24}$$

where v_{cs} is the velocity of the current sheet measured with the magnetic probes. Included is an IDL code, *rundown_vel.pro* in Sec. C.2.2, which calculates velocity and mass of the current sheet.

Current Density Measurements

The radial current density between two axial probes is proportional to the difference of the magnetic field if the magnetic field is a function of r and z only. If the current is azimuthally

symmetric, the magnetic field at each probe is given by

$$B_\theta = \frac{\mu_0 I_{\rm enc}}{2\pi r_{\rm wall}} \tag{5.25}$$

The only way for $I_{\rm enc}$ to be different between two probes is for part of the current to attach to the wall. The average radial current density, J_r, is then given by

$$\begin{aligned} J_r &= \frac{I^{(2)} - I^{(1)}}{2\pi r_{\rm wall} \triangle z} \\ &= \frac{B_\theta^{(2)} - B_\theta^{(1)}}{\mu_0 \triangle z} \end{aligned} \tag{5.26}$$

where $B_\theta^{(i)}$ is the azimuthal magnetic field at location i and $\triangle z$ is the axial spacing of the probes. This analysis is only valid in the acceleration region where kink modes in the current are not seen. Another method of looking at the axial variation of magnetic field is to plot the percentage of the total plasma current, $I\%$ which makes it past the probe

$$I\% = \frac{B_\theta^{(i)}}{B_\theta^{(-120)}} \tag{5.27}$$

where $B_\theta^{(-120)}$ is used instead of the plasma current because of the lack of frequency compensation. This method allows one to monitor the current density in the accelerator, which is proportional to $\partial I\%/\partial z$, and look for axial perturbations in the Z-pinch, shown in Fig. 5.9. The horizontal line is placed at the end of the acceleration region. Below this line the current density along the outer electrode is given by Eq. 5.26. The contours in this region may also be thought of attachment locations of the current in the acceleration region. Above the line all of the current is assumed to be in the Z-pinch, as discussed in Sec. 5.5.4. The contours in this region are assumed to be from the radial motion of the Z-pinch. The contours in this region after 40 μs are from the axial motion of a kink.

5.6 Interferometry

A standard method of measuring the electron density, n_e, is with interferometry. The index of refraction of a plasma is proportional to the line integrated density. ZaP uses a two chord Mach-Zehnder interferometer inherited from the High Beta Q machine [53]. Improvements have been made to the hardware and data analysis routines.

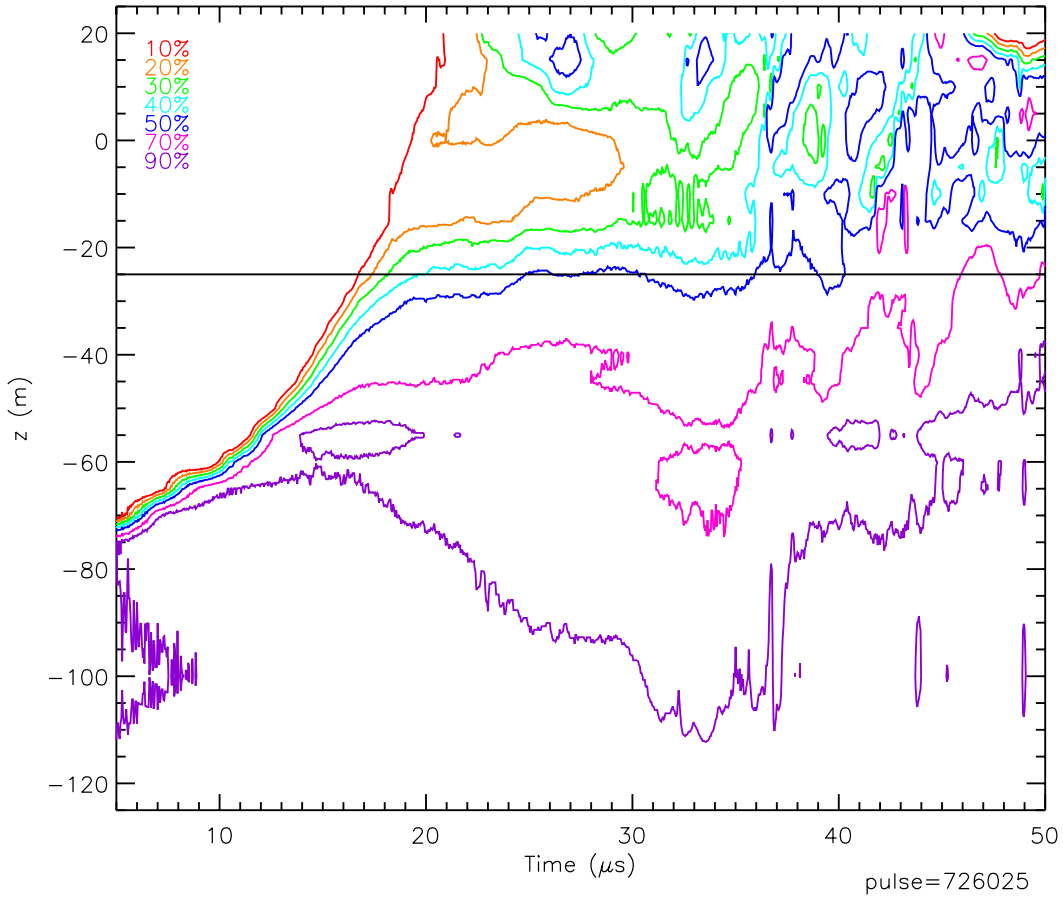

Figure 5.9: The magnetic data normalized by the current showing the location of the current in the accelerator and perturbations in the Z-pinch. The horizontal line is located at the end of the acceleration region. The contour below this line are the axial attachment locations of the radial current. Above this line, the contours show the location of perturbations in the plasma.

5.6.1 Theory

Interferometry measures the index of refraction of a medium. The dispersion of relation of a uniform cold plasma is governed by the Appleton-Hartree relationship.

$$N^2 = 1 - \frac{X(1-X)}{1 - X - \frac{1}{2}Y^2\sin^2\theta \pm [(\frac{1}{2}Y^2\sin^2\theta)^2 + (1-X)^2Y^2\cos^2\theta]^{1/2}}$$
$$X = \frac{\omega_p^2}{\omega^2}, Y = \frac{\omega_c}{\omega}, N = \frac{kc}{\omega}, \omega_c = \frac{eB_0}{m_e}, \omega_p = \left(\frac{n_e e^2}{\epsilon_0 m_e}\right)^{1/2} \tag{5.28}$$

where θ is the angle between the magnetic field and wave, ω is the frequency of the wave, ω_p is the plasma frequency, ω_c is the cyclotron frequency, k is the wave number, B_0 is the magnitude of the magnetic field, N is the index of refraction, and n_e is the electron number density. For a magnetic wave propagating perpendicular to the magnetic field the Appleton-Hartree relationship reduces to an ordinary wave.

$$N^2 = 1 - X = 1 - \frac{\omega_p^2}{\omega} \tag{5.29}$$

This defines the index of refraction for a cold plasma.

An interferometer splits the beam of a laser into two beams. The scene beam is run though the medium. The reference beam is run the same length as the scene beam around the medium. The beams are then recombined. The difference in the index of refraction that both beams see causes a phase difference, $\triangle\phi$, in the light is given by

$$\triangle\phi = \int (k_{medium} - k_0) dl = \int (N-1)\frac{\omega}{c} dl \tag{5.30}$$

where k_{medium}, k_0 are wave numbers in the medium and air. For light propagating perpendicular to the magnetic field the phase lag reduces to

$$\triangle\phi = \frac{\omega}{c} \int \left[\left(1 - \frac{n_e}{n_c} \right)^{1/2} - 1 \right] dl \tag{5.31}$$

$$n_c \equiv \frac{\omega^2 m \epsilon_0}{e^2} \tag{5.32}$$

where n_c is the cutoff density. When the n_e is small compared to n_c the phase shift reduces to

$$\triangle\phi = \frac{\omega}{2cn_c} \int n_e dl \tag{5.33}$$

The change in phase is proportional to the line integrated density.

ZaP uses a heterodyne, quadrature, Mach-Zehnder interferometer. A heterodyne interferometer uses light with two different frequencies. When the beams are combined, a beat at the difference of the frequency is created. Electronics at RF frequencies can then be used. The beat is then compared to a reference beat. The phase between the beats is the quantity of interest. By recording the phase in quadrature, the phase and the phase plus

$\pi/2$, information is not lost when the phase goes through zero. Accurate plasma density measurements are made with this technique [32, 8].

The system uses a Bragg cell, an acousto-optic modulator, to split the laser beam into two beams with different frequencies. The unmodulated scene beam is passed through the plasma. The modulated reference beam is directed around the chamber. The lengths of both beams are keep the same to avoid problems due to the finite coherence length of the laser. Similar optical components are used with each beam, keeping the plasma as the only difference between the beams. The beams are then recombined with a beam splitter. The recombined beam must be coaxial to ensure constructive and destructive interference of the light from both beams across the detector. The intensity of the combined beam is given by

$$I = A^2 + B^2 + 2AB\cos(2\pi f_{\text{modulator}} t + \triangle\phi) \tag{5.34}$$

where A and B are the intensity of the reference of scene and reference beams, and $f_{\text{modualater}}$ is the modulator frequency.

5.6.2 Hardware

Most of the hardware is from the interferometer used on the High Beta Q machine . The laser has been replaced with a 7 mW HeNe laser, $\lambda = 632.8$ nm. A 40 MHz oscillator is used to drive the Bragg cell. Half of the power from the oscillator is used as the reference signal in the quadrature detectors. The Bragg cell splits the HeNe beam into two beams. The reference beam has a 40 MHz shift in its frequency. Two chords are obtained by using a 50/50 beam splitter to divide the reference and scene beams into two beams each [29]. Mirrors then guide the scene beams through various ports of the vacuum chamber, labeled in Fig. 4.1. The reference beams are directed beside the scene beam and over the chamber. A reference and a scene beam pair are then combined with a beam splitter. The light is then focused onto a photo detector. Vibrations are keep to a minimum by mounting all the optics on two optical tables on either side of the vacuum chamber. The tables sit on rubber to dampen the vibrations.

The signals are then converted to $\sin\triangle\phi$ and $\cos\triangle\phi$ by an analog circuit. The photodetectors are biased with -90 V with the circuit shown in Fig. 5.4. The signal from the

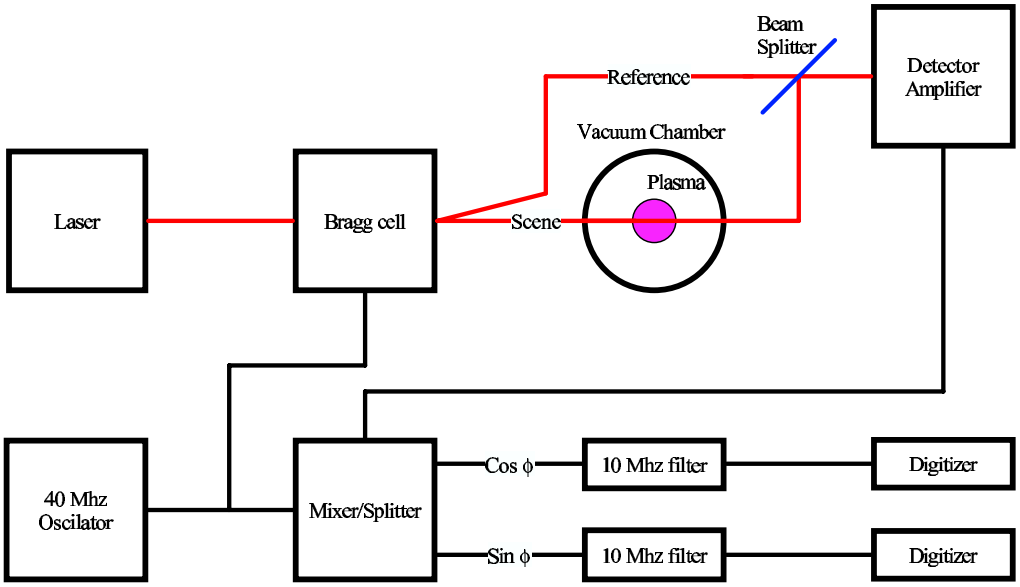

Figure 5.10: A block diagram showing the major components of the interferometry system. A 40 MHz oscillator is used to drive the Bragg cell and provides the reference signal to the mixer/splitter boxes. The unshifted laser beam is used as the scene beam. The shifted laser beam is used as the reference beam. A 40 MHz band pass amplifier, IF amplifier, is used near the detectors. Low pass filters are used before the digitizers to remove the beat frequency from the signals.

photodetector is amplified. The signals from the photodetectors and the oscillator are combined in UW quadrature detectors [53]. The detectors output the sine and cosine of the phase between the two signals. The 40 MHz beat is removed with a 10 MHz low pass filter. The $\sin \triangle\phi$ and $\cos \triangle\phi$ are then digitized. The block diagram of the setup is shown in Fig. 5.10.

The photodetectors and IF amplifiers are housed in a shielded box. During a pulse, the interferometer picks up electronic noise. To alleviate this problem an electrostatic shield was place around the hot and cold plates. The noise during breakdown has been significantly reduced. Electrostatic noise during the compression of the pinch has been reduced by grounding power supplies for the IF amplifiers at the shielded box. The ground loop introduced by connecting the grounds has not caused any problems since all of the

leads are inside the copper braid which grounds the shielded box.

5.6.3 Data Analysis

The sine and cosine of the phase are converted to densities with an IDL routine, *dense_fin.pro* Sec. C.3.1. The digitized data, x and y is given by the circle

$$\begin{aligned} x &= R\cos\phi(t) + x_0 \\ y &= R\sin\phi(t) + y_0 \end{aligned} \tag{5.35}$$

where R is the radius of the circle, $\phi(t)$ is the phase, and (x_0, y_0) the centers of the lissajou. The center of the circle is found by blocking the beams and recording the signals from the quadrature detectors setting the value R to 0. x_0 and y_0 can then be found. A least squares fit was tried for the calculation of the offsets [53]. The method does not work when the radius of the circle is a function of time as shown in Fig. 5.11. The noise which causes the radius to change has been reduced on the signals. The least squares fit still has problems finding the center when only a small arc of circle is recorded on a pulse. The measured offsets are subtracted from x and y, moving the center of the coordinate system to the center of the lissajou. The resulting signals are $R\sin\phi(t)$ and $R\cos\phi(t)$

The phase is found with $\phi(t) = arctan(\sin\phi(t), \cos\phi(t))$, arctan is an IDL routine which returns ϕ from π to $-\pi$. The vibrations in the system cause the initial phase to be arbritrary. A rotation of the coordinate system is done to set the initial phase to zero. To reduce the high frequency noise of the $\sin\phi(t)$ and $\cos\phi(t)$ data a double smooth was applied to the data. High frequency noise is removed from the data by smoothing and written to the tree as a separate signal. The method described in Sec. 5.1.3, with $f_{\text{filter}} = 10$ MHz, is presently being used to remove the noise from the interferometry data. The smoothed and unsmoothed data were analyzed.

Vibrations in the optical system also cause a change in $\phi(t)$. The time scale of the vibrations is much longer than any of the plasma time scales. The change in $\phi(t)$ due to vibrations, $\phi_{vib}(t)$, during a pulse is assumed to be linear. A line is fit through $\phi(t)$ early in time. As the length of the plasma pulse has increased, or if the mirrors are at the maximum displacement, the linear assumption is no longer valid. Interferometry data should be taken

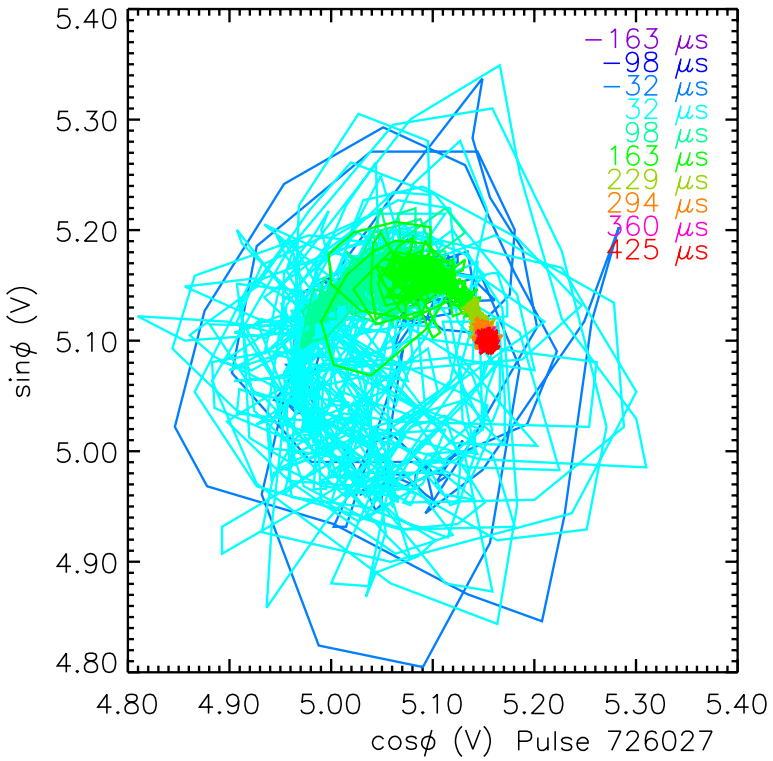

Figure 5.11: The lissajou from pulse 726027 shows the radius varying with time. The start time of each color is shown in the upper left corner. The noise from the ignitron triggering can be seen in the blue trace. Electrostatic noise during the plasma pulse causes the radius to be a function of time as shown in the cyan trace. The small changes in the red trace show that the vibrations in the system are small.

long after the plasma pulse to measure $\phi_{vib}(t)$. Since $\phi_{vib}(t)$ can go through $\pm\pi$ late in time, the fringe jump routine discussed in the next paragraph is used for the data after the plasma pulse. A fourth order polynomial is then used for $\phi_{vib}(t)$. The fourth order polynomial has been used after September, 2003. This is then subtracted from ϕ leaving only changes from the plasma.

$$\phi_{plasma}(t) = \phi(t) - \phi_{vib}(t) \tag{5.36}$$

where $\phi_{vib}(t)$ is the change in $\phi(t)$ due to vibrations and $\phi_{plasma}(t)$ is the phase change from the electron density.

When the $\phi_{plasma}(t)$ is close to $\pm\pi$, a small change in the angle results in a change of

$\phi(t)$ of approximately $\pm 2\pi$. This is referred to as a fringe jump. Phase changes are required to be less than π during one digitizer time step. This sets the maximum frequency of the interferometer to the Nyquist frequency. Any changes in the phase during one time step greater than this are assumed to be caused from a fringe jump. When there is a fringe jump, 2π is added to, or subtracted from, $\phi_{plasma}(t)$ depending on the direction of the jump. The rotation of the coordinate system reduces the number of fringe jumps which must be found.

Diffraction, noise in the system, or other light sources can cause the radius of the signal to go to zero. When this occurs the phase information is often lost. The density at the end of the plasma pulse often does not return to zero when this occurs. This problem is solved by running the fringe jump algorithm forward and backward in time. The two phases, ϕ_{fore} and ϕ_{back}, are then compared to each other. When they match, the phase information was followed during the pulse and no corrections are made. When ϕ_{back} is greater than ϕ_{fore}, 2π is added ϕ from the earliest time when $\phi_{\text{fore}} - \phi_{\text{back}}$ is positive. $\phi_{plasma}(t)$ is then multiplied by a conversion factor to get the line integrated density. The average density is calculated assuming it is uniform across the acceleration region or inside of the pinch. The densities are then written to the tree.

Interferometry Abel Inversions

When both chords of the interferometer are located at the same z location the local plasma density can be found with an Abel inversion [27]. When the number density, $n(r)$, is a function of r only, chord integrated measurements, $N(y)$, are given by

$$\begin{aligned} N(y) = & \int_{-\infty}^{\infty} n(r)dx \\ & = 2\int_{y}^{\infty} n(r)\frac{rdr}{\sqrt{r^2 - y^2}} \end{aligned} \tag{5.37}$$

where a description of the variables is given in Fig 5.12. Abel solved Eq. 5.37

$$n(r) = -\frac{1}{\pi}\int_{r}^{\infty} \frac{N'(y)}{\sqrt{y^2 - r^2}}dy \tag{5.38}$$

where $N'(y)$ is the derivative of N with respect to y. Since the integral goes to infinity, at least one chord is needed outside of the plasma to measure when $N(y)$ and $N'(y)$ are zero.

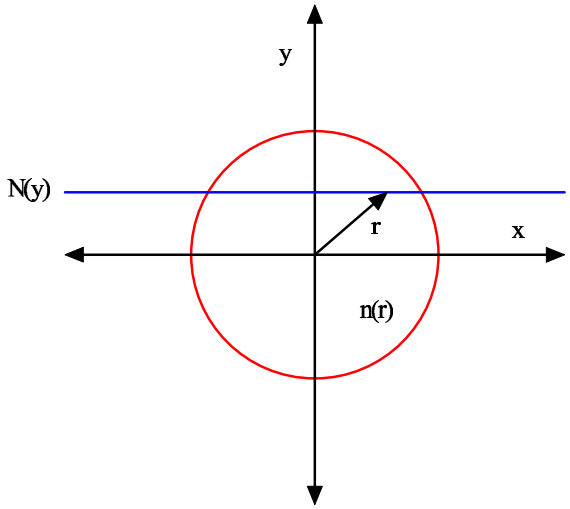

Figure 5.12: Line integrated data is deconvolved with an Abel inversion. The plasma density, $n(r)$, is a function of r. A chord of interferometry measures the line integrated density, $\int n(r)dl$ along the blue line.

When the edge of the plasma is unknown, assumptions about the edge region must be used to determine the radial density profile.

A simple model is used to estimate the average density in the pinch when two chords are located at the same z location. A simple Abel inversion is used where the density is assumed to be piecewise constant in each shell. The two regions are shown Fig. 5.13 The measured line integrated densities, N_1 and N_2, are given by

$$\begin{bmatrix} N_1 \\ N_2 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} \\ 0 & L_{22} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{5.39}$$

The average density of the background and Z-pinch plasma are found multiplying the line integrated densities by the inverse of the L matrix. This method is only valid when the plasma is centered in the outer electrode. The ideas used in this inversion are expanded in Sec. 5.7.5, where four parameters vary between each shell.

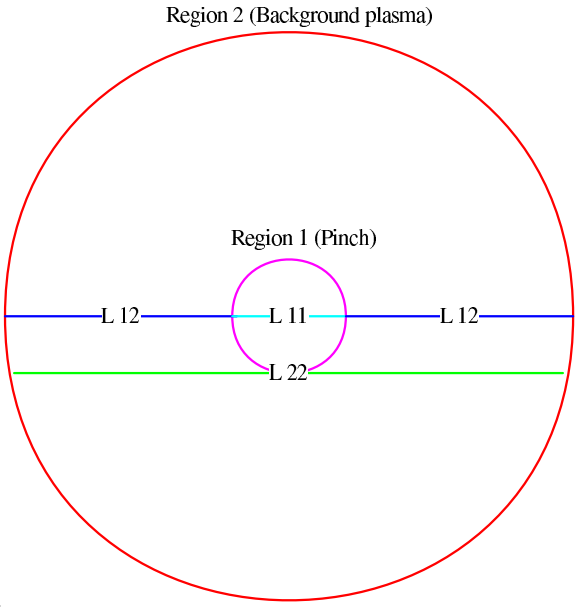

Figure 5.13: The assembly region is divided into two sections. The plasma density is assumed to be piecewise constant in each section. Chord 2 (green) only passes through the outer region. Chord 1 (blue and cyan) passes through both regions. The average density in each region are found using Eq. 5.13

5.7 Spectroscopy

On ZaP, passive spectroscopy is used to measure impurities, temperatures, and velocities in the plasma. When the ion-ion equipartition time is small, the bulk plasma and the impurities have the same properties [67]. The impurities emit line radiation with a spectral intensity dependant on the characteristics of the impurities. The plasma parameters can be calculated from the shape of the spectral intensity. Section 5.7.1 discusses the processes which affect the shape of the spectral intensities.

Each spectrometer has its own unique character [46]. Three different spectrometers are used to measure the spectral intensities on the ZaP project. The Jarrel Ash, CCD, spectrometer measures both the time integrated spectral intensity through one chord of the plasma over the entire pulse and the time evolution of a narrow region of the spectra. Initial measurements of the plasma parameters using this spectrometer were used to design the

ICCD spectrometer. The ion doppler spectrometer, IDS, measures the time evolution of intensity for 16 wavelengths from one chord through the plasma. This spectrometer was on loan from the Himeji Institute of Technology for two months [41]. The Acton Research Corporation spectrometer with image intensified CCD, ICCD, measures the spectral intensity of 20 chords though the plasma over a small time interval of the plasma pulse. The ICCD spectrometer is used to measure radial profiles of the plasma parameters. Sections 5.7.2 to 5.7.4 describe the hardware, calibration and data analysis of each spectrometer.

Since the spectral intensities are chord averaged, a deconvolution is required to calculate the local plasma parameters [20]. The process described in Sec. 5.7.1 cannot be simply reversed. Section 5.7.5 describes the deconvolution technique used to calculate the local plasma parameters in ZaP. This technique calculates the local plasma parameters and locates the center of the Z-pinch based on a shell model.

5.7.1 Theory

Despite all of the precautions taken to keep the vacuum system clean, impurities, C, O, *etc.*, are present in the plasma are typically partially ionized. Their ionization state, when plasma is in coronal equilibrium, is described in Carolan's paper [12]. The impurities radiate light with a known wavelength. The wavelength of the light is shifted due to the ion's motion. Useful information is gained by measuring the details of the radiation from the impurities.

Passive spectroscopy uses light emitted from the impurities in the plasma to measure various parameters. Bound electrons in the plasma are excited to higher levels by collisions with electrons. When the electrons fall to a lower state a photon is emitted. The frequency and wavelength of the photon from each transition is invariant in the particle's frame of reference given by Eq. 5.6. The wavelengths for various ions can be found online in the NIST Atomic Spectra Database [38]. The abundance of impurities in the plasma can be measured by monitoring the relative intensities of different ions and ionization states. The relative intensity measurements are used as a qualitative measurement on the ZaP experiment.

The wavelength of the emitted photons is different in the viewing frame. The velocity difference, v_i, between the particle's and the viewing frames of reference Doppler shifts the

wavelength by

$$\lambda_{\text{measured}} = \lambda_0 \left(1 - \cos\theta \frac{v_i}{c}\right) \tag{5.40}$$

where λ_{mesured} is the measured wavelength, λ_0 is the emitted wavelength, and θ is the angle between the velocity vector and viewing chord. Since the emitting particles have a thermal distribution, the components of the velocity along the viewing chord will have a shape given by distribution function, $f(\mathbf{v})$. When the ions are in thermal equilibrium, $f(\mathbf{v})$ is given by

$$f(\mathbf{v}) = \left(\frac{M_i}{2\pi k T_i}\right)^{\frac{3}{2}} \exp\left[-\frac{M_i(\mathbf{v}-\mathbf{v}_i)^2}{2kT_i}\right] \tag{5.41}$$

where M_i is the ion mass, T_i is the ion temperature, and $\mathbf{v}_i$ is the average ion velocity. The local spectral emissivity, $e(\lambda)$ due to the Doppler effect is given by

$$e(\lambda) = \frac{E}{\sqrt{2\pi}\sqrt{\frac{kT_i\lambda_0^2}{m_i c^2}}} \exp\left[-\frac{(\lambda - \lambda_0 + \mathbf{s}\cdot\mathbf{v}_i\lambda_0/c)^2}{2\,\frac{kT_i\lambda_0^2}{m_i c^2}}\right] + Br \tag{5.42}$$

where E is the emissivity, T_i is the ion temperature, m_i is the ion mass, $\mathbf{s}$ is a unit vector along the viewing chord, $\mathbf{v}_i$ is the ion velocity, and Br is the level of the background radiation. The difference between the wavelength of the centroid of the spectral intensity and the emitted wavelength is related to the average speed of each species with Eq. 5.40. The full width half max, FWHM, of the spectral intensity, $\triangle\lambda_{FWHM}$, is related to the temperature with

$$\begin{aligned} \triangle\lambda_{FWHM} &= 2\lambda_0\sqrt{2\ln(2)\frac{kT_i}{M_i c^2}} \\ &= 7.71\times10^{-5}\lambda_0\sqrt{\frac{T_i}{M_i}} \end{aligned} \tag{5.43}$$

where T_i is in eV and M_i is in AMU.

The local spectral intensity shape can be affected by mechanisms other then Doppler effects. Fine structure, pressure or Stark broadening, and Zeeman splitting are some of the other mechanisms. Fine structure broadening is due to the internal structure of the ion which leads to uncertainties in the energy levels. Fine structure broadening leads to a widening and skewing of the line emission. This effect is much smaller than the doppler broadening of the lines used for temperature measurements. The C IV line at 465.83 nm

is widened by fine structure broadening. This line is only used for velocity measurements. Pressure or Stark broadening is due to the electric fields from plasma electrons and ions. Pressure broadening changes the line shape to a Lorentzian given by

$$I(\lambda) = I(\lambda_0)\frac{1}{1+\left[(\lambda_0-\lambda)\,2\pi c\tau/\lambda_0^2\right]^2} \tag{5.44}$$

where τ is the lifetime of the higher energy state. This effect is small for the lines used. The magnetic fields splits the line according to the Zeeman effect

$$\triangle\lambda = \frac{gM\mu_B B\lambda_o^2}{\hbar c} \tag{5.45}$$

The lines used on ZaP are not significantly broadened by these mechanisms.

The spectral intensities are measured along a chord through the plasma. The measured spectra are chord integrated and can be affected by the optical depth, τ, of the plasma. The spectral intensity of measured from a collection point through a chord in the plasma is given by [21]

$$I(\lambda) = \int_0^{\tau_a} I_t(\lambda,\tau)e^{-\tau}d\tau \tag{5.46}$$

where $I_t(\lambda,\tau)$ is the spectral intensity at location τ. When the plasma parameters do not change, the optical depth is given by

$$\tau(\lambda) = 1.5\times10^{-9}\lambda n_{\text{lower}} l\sqrt{\frac{\mu E_H}{kT}} \tag{5.47}$$

where n_{lower} is the number of ions in the lower energy state, l is the smallest dimension of the plasma, μ is the atomic mass and E_H is the excitation energy. The optical depth is difficult to measure when the temperature varies along the chord and the number of ions in the lower energy state is unknown. Integrating Eq. 5.46 by parts shows that the maximum intensity of the plasma is governed by the plasma's black body radiation level. The intensity, I, given by

$$I(\lambda) = \frac{2\pi hc^2}{\lambda^5}\frac{1}{\exp\left(\frac{hc}{\lambda kT}\right)-1} \tag{5.48}$$

where T is the temperature at $\tau \approx 1$. When measured intensity approaches the blackbody level, shown in Fig. 5.14, optical thickness effects become important. When the intensity of

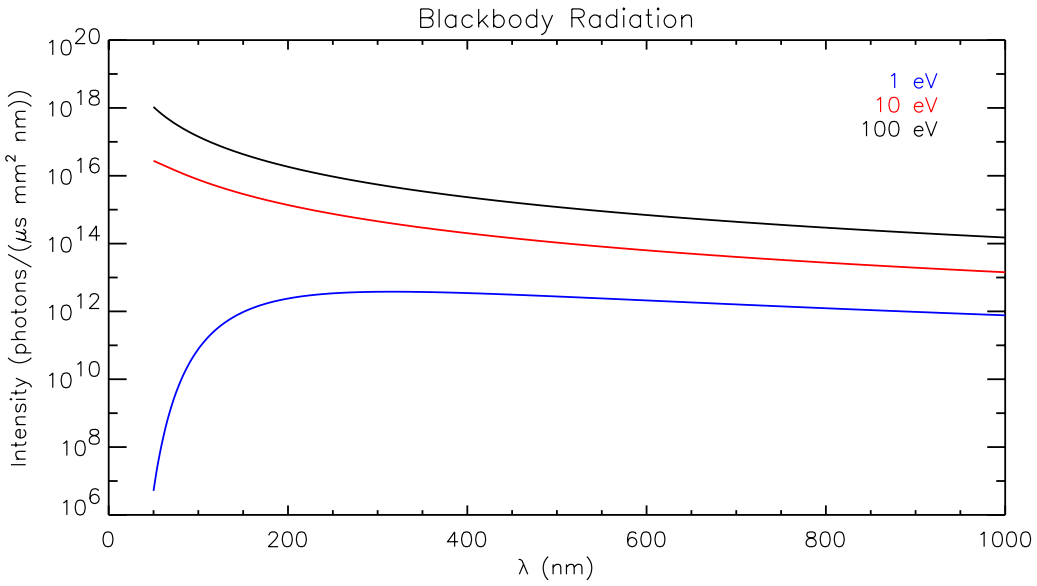

Figure 5.14: The spectral intensities from black body radiation are calculated for three temperatures. The number of photons measured by the experiment does not approach these levels.

the measured light is small compared to the blackbody level, the absorption of light can be ignored.

The plasma is optically thin from the wavelength region used in ZaP. An optically thin plasma simplifies the calculation of the plasma parameters. When the plasma is cylindrically symmetric, the intensity of each wavelength of light is given by Eq. 5.37. The chord-integrated emissivity and velocity are found by fitting Gaussians to the measured spectral intensities. These parameters are calculated with the assumption that the plasma parameters are uniform along the chord.

The radiation is resolved into its spectral components using spectrometers. There are many spectrometer configurations [46]. The two configurations used in ZaP spectrometers are the Fastie-Elbert and Czerny-Turner. The Fastie-Elbert consists of a large spherical mirror and a plane diffraction grating. The Czerny-Turner configuration uses two concave mirrors and a piano grating. The spectrometers convolve the spectra with an instrument function, $I_{\text{instrument}}$, the spectral intensity measured by the spectrometer of a delta function.

Another term for the instrument function is the spectral response of the spectrometer. The measured spectral intensity, M_i is given by

$$M_i(\lambda) = \int_{-\triangle\lambda}^{\triangle\lambda} I(\lambda - \lambda') I_{\text{instrument}}(\lambda') d\lambda' \tag{5.49}$$

where $\triangle\lambda$ is the edge of the instrument function. The calculation of the plasma parameters from the measured spectral intensities is discussed in each of the following sections.

5.7.2 CCD spectrometer

The first plasma made in the ZaP experiment was a He glow discharge. The spectra from this discharge was recorded with the CCD spectrometer. The spectrometer is both a survey spectrometer and monochrometer. The initial spectrum of the plasma was taken with the CCD spectrometer. The results from the CCD spectrometer have been used to design the ICCD spectrometer.

Hardware

Light from the plasma is gathered from one of two telescopes. The telescopes are focused at the center of the machine, 100 mm (90 degree view) and 200 mm (35 degree view). Each telescope is mounted on the appropriate optical port on the bottom of the machine which view the Z-pinch at z=0. A fused silica fiber can be placed at seven spatial locations to view different chords of the plasma. A telescope with an infinite focal length has also been used. A ferroelectric liquid crystal, FLC, shutter can be placed inside this telescope. The shutter can be gated down to 10 μs for time history studies. The shutter does not pass light in the UV region.

The CCD spectrometer is 0.5 m Fastie-Elbert spectrometer. Three interchangeable gratings, a fine (2360 g/mm blazed at 240 nm), medium (a 1180 g/mm blazed at 300 nm), and a coarse (147.5 g/mm blazed at 300 nm) grating, have been used. The CCD spectrometer has two exit planes. A CCD camera is located at one exit plane. The other exit plane has a slit with a PMT behind it. The grating is rotated with a sine wave bar to center different lines on the CCD spectrometer. Light from the grating is imaged on to a CCD. The minimum exposure of the CCD spectrometer is 0.01 s, longer than the plasma lasts. A

swing out mirror is used to monitor the time evolution of one line on the CCD spectrometer. The mirror has been replaced with a 50/50 beam splitter. Both the relative intensities of a number of lines and the time evolution of one of those lines are measured every shot with one instrument. An added benefit is the nonrepeatability associated with moving the swing out mirror has been eliminated.

Calibration

The CCD spectrometer has a large amount of backlash in the wavelength adjustment dial. It requires a wavelength calibration every time the dial is moved. Normally two lines from one or two of the calibration lamps are present in the wavelength region. This allows the use of the calibration feature of the Kestrel Spec control and acquisition software to calculate the pixel to wavelength conversion. When two lines are not present, the spectrometer is moved to a nearby region where two lines can be measured or known lines in a plasma pulse can be used to get the wavelength calibration. With the coarse and medium gratings this process could be used for all dial settings. The instrument function is the spectral intensity from one of the calibration lines.

The calibration of the fine grating was more involved because the grating is rotated when the dial is set to zero. Typically one calibration line can be seen on the fine grating. Spectra of each calibration lamp was acquired from dial setting 2000 to 9200 in increments of 100. Two successive dial settings, *dial*, were over plotted. Then a constant was added to the x axis of the higher dial setting until similar peaks lined up with each other, shown in Fig. 5.15. This was repeated for all dial settings with each calibration lamp. The constants *versus* dial settings were then plotted, and a line was fit to the data.

$$cons = 4.96548 + 0.00808dial \ . \tag{5.50}$$

The start pixel for each data set was then set to

$$pixel(dial) = \sum_{i=0}^{\frac{dial-2000}{100}} 4.96548 + 0.00808(2000 + 100i) \ . \tag{5.51}$$

The relationship between wavelength and pixel is now known. From this the dispersion and

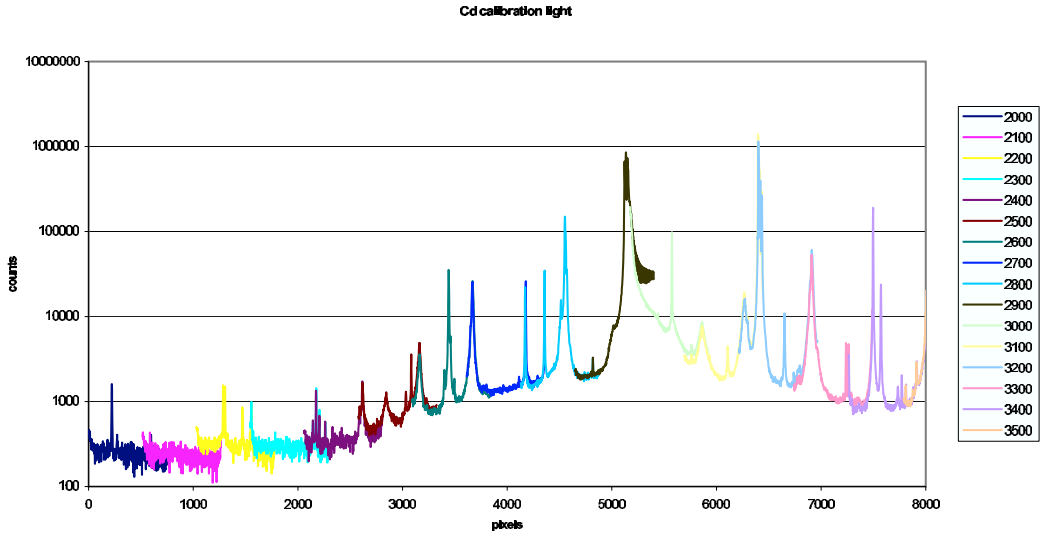

Figure 5.15: Peaks in spectra from the CCD spectrometer fine grating calibration are aligned by adjusting the x axis. The x axis is in pixels. A constant has been added to the x axis of successive dial settings until the line emission agrees in pixel space.

center wavelength as a function of pixel are found.

$$\begin{aligned} pixel_{center}(dial) &= pixel(dial) + 375 \\ \lambda_{cen} &= 209 + 0.00898 pixel_{center} - 6.82 \times 10^{-09} pixel^2_{center} \\ \frac{d\lambda}{dpixel} &= 0.00898 - 1.364 \times 10^{-08} pixel_{center} \end{aligned} \tag{5.52}$$

The center wavelength, dispersion and span were then plotted *versus* dial setting, shown in Fig. 5.16. These charts are used to setup the CCD spectrometer. The final wavelength calibration must be done with calibration lamps as described above because of the none repeatability of the dial.

The PMT is aligned with a calibration lamp and an ammeter. The grating was rotated until the largest current was seen. When the mirror was used, it was moved and a spectra was taken on the CCD. The pixel of the maximum signal was recorded. During pulses the desired line was set to that pixel. The mirror was moved into path of the light and data were taken on the PMT. The mirror has been replaced with a beam splitter which allows the CCD and PMT to both be used during a shot. The alignment of the PMT

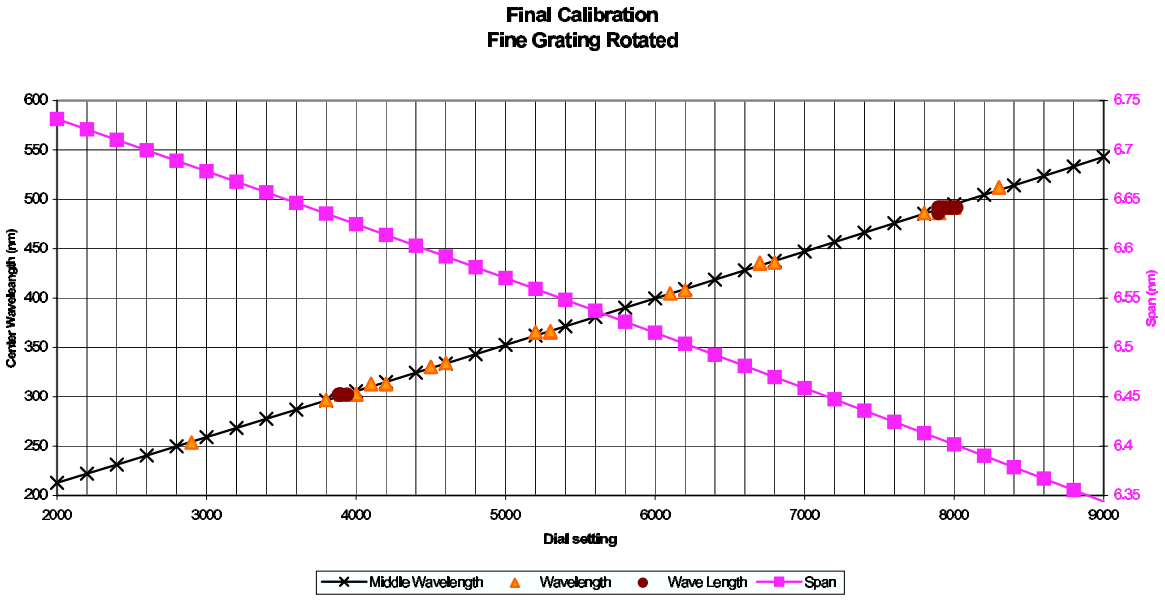

Figure 5.16: Shown are final calibration curves for the CCD spectrometer with the fine grating. The black line shows the center wavelength verse dial setting and the magenta line is the dispersion verses dial setting.

center wavelength on the CCD is well known because the beam splitter is not moved before recording the spectra.

Data Analysis

Presently, the main use of the CCD spectrometer is line identification, impurity control, and the time evolution of various ions. Data from the CCD spectrometer with the shutter are used to find the plasma parameters with the method discussed above. These were used to design the ICCD spectrometer. The CCD spectrometer does not have the spatial nor temporal resolution of the ICCD spectrometer. Impurity lines are identified by comparing the relative intensity measured line to the relative intensities found in the NIST database [38]. When the measured relative intensities are proportional to the relative intensities from the database, the ion is present in the plasma. Surveys of the impurity line radiation show the main impurities are N, O, and C, see Fig. 5.17. Relative intensities between different

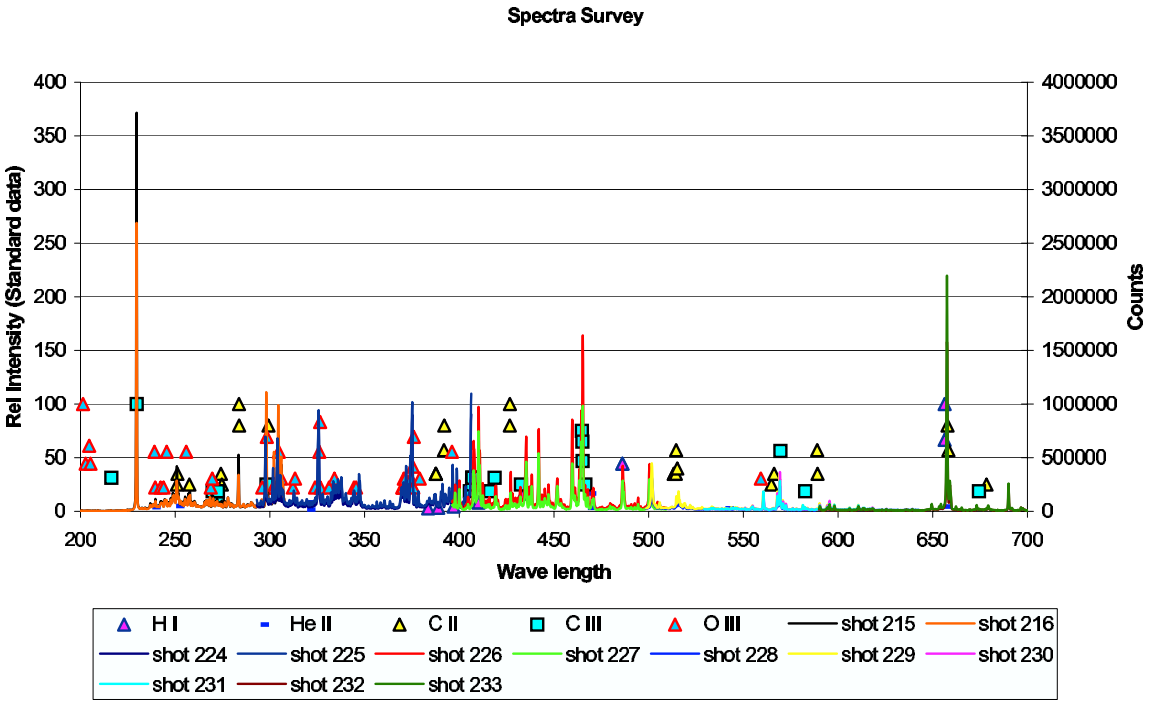

Figure 5.17: The initial spectrum of the Z-pinch was taken with the CCD spectrometer. The measured spectra from similar pulses are shown as solid lines. Impurity lines with scaled relative intensities are shown as symbols. The level of the metal lines is small over the entire plasma pulse.

ions are used as a qualitative measurement of impurity measurement. The intensity of the line is proportional to the density of the ion. When the intensity of the C III line changes between shots the average density of C III has changed accordingly. The metal content of the plasma is monitored throughout a run day by comparing the intensity of the discrete line radiation of useful impurities to the intensity of the "forest" lines from the metals. When the intensity of the metal lines gets large, changes in the experiment setup are made which decrease the metal content in the plasma.

Data from the PMT are used to measure the time evolution the impurity radiation from one ion. Often the CCD spectrometer and ICCD spectrometer monitor the same line. The CCD spectrometer provides information on the repeatability of the ion's evolution between pulses. It also provides a means to find the time window of the ion's radiation. The time

of peak emission and burn through are compared to coronal equilibrium [12].

5.7.3 IDS

The IDS spectrometer measures the time evolution of the intensity of 16 wavelengths through one chord of the plasma. An array of 16 PMTs is used to measure the intensities. The calibration of the IDS spectrometer is done by sweeping a line across the PMT array. Noise is picked up on the IDS spectrometer during voltage spikes. A data analysis routine is used to evaluate the noise and calculate the plasma parameters.

Hardware

The IDS spectrometer was used on the Helicity Injected Torus experiment [41]. Light for the IDS spectrometer was collected with the CCD spectrometer telescopes with slight modifications made to hold the fibers. The spectrometer is a MC-100: Diffraction Grating Spectrometer. This is a 1.0 m, Czerny-Turner spectrometer with a 1800 grooves/mm grating blazed at 300 nm. Although the spectrometer has a wavelength range from 250 nm to 700 nm, measurements of the C III line at 229 nm have been made with this system. A cylindrical lens is placed at the exit slit to increase the dispersion of the light. A 16 anode PMT on a translator is used to measure the intensity of light in 16 wavelength regions. The PMT is biased with 550 to 850 volts and the signals are terminated with 1 kΩ at the digitizer.

Calibration

The IDS spectrometer can only measure one line at a time. The dispersion, relative intensity, and instrument function are found by moving a known line across the PMT array at a known rate and the voltages from the array are digitized. Various lines from a mercury lamp were swept across the PMT array at 1 nm per second.

Since the light from the Hg calibration lamp is dimmer than the plasma, the termination of the signals must be changed to 220 kΩ. The fastest speed on the grating motor is one nanometer a second. At this speed a line is swept across all 16 channels in 0.5 seconds. The

digitizers are set to their slowest rate to acquire all of the lines. A slower speed will increase the number of peaks from the lamp increasing the accuracy of the calibration.

Acquisition of the calibration curves was aided by connecting a scope to channel 1 and 16. Using the motor control in reverse, the line of interest is moved past channel one. The motor was started forward sweeping the line across the array and data acquisition was triggered acquiring the curve. The spectrometer is sensitive to the entrance slit conditions, wavelength, exit slit conditions and PMT location. Calibration curves for all of the operating conditions were obtained.

The calibration lamp output a rectified 60 Hz sine wave. The FWHM of the instrument function is equal to the dispersion between channels, shown in Fig. 5.18. A spline fit was done through the local peaks of the data to calculate the amplitude, FWHM, and time of each peak for each channel of the PMT, shown in Fig. 5.19. The peak voltages were also fit to a Gaussian which compares well to the spline fit. The values from the spline fit were then used to calculate the instrument temperature, relative intensities and dispersion of the spectrometer. An IDL routine, *IDS_CAL4.pro* Sec. C.4.1, was used to calculate the calibration parameters. The instrument temperature is a function of slit width. The dispersion and relative intensity are constant over the operating parameters.

The difference between the dial and the wavelength measured at channel 8 was found with the aid of an ammeter. The dial was set below a known wavelength of the calibration lamp. The ammeter was connected to channel eight. The fiber was then illuminated with a mercury lamp. The dial was increased until the maximum signal was seen on channel. The difference between the dial and the known wavelength is about 0.3 nm.

The dial did not have the desired accuracy for velocity measurements. With the 35° viewing angle a 0.01 nm shift corresponds to a velocity of 9156 m/s. To alleviate this problem, only the fiber was moved during a survey. A correction factor, λ_{dial}, which set the average velocity to zero during the quiescent period of the m_1 mode on all 90° pulses was found. During this time interval the radial velocity of the Z-pinch is zero. This factor was added to the wavelength until the dial was moved.

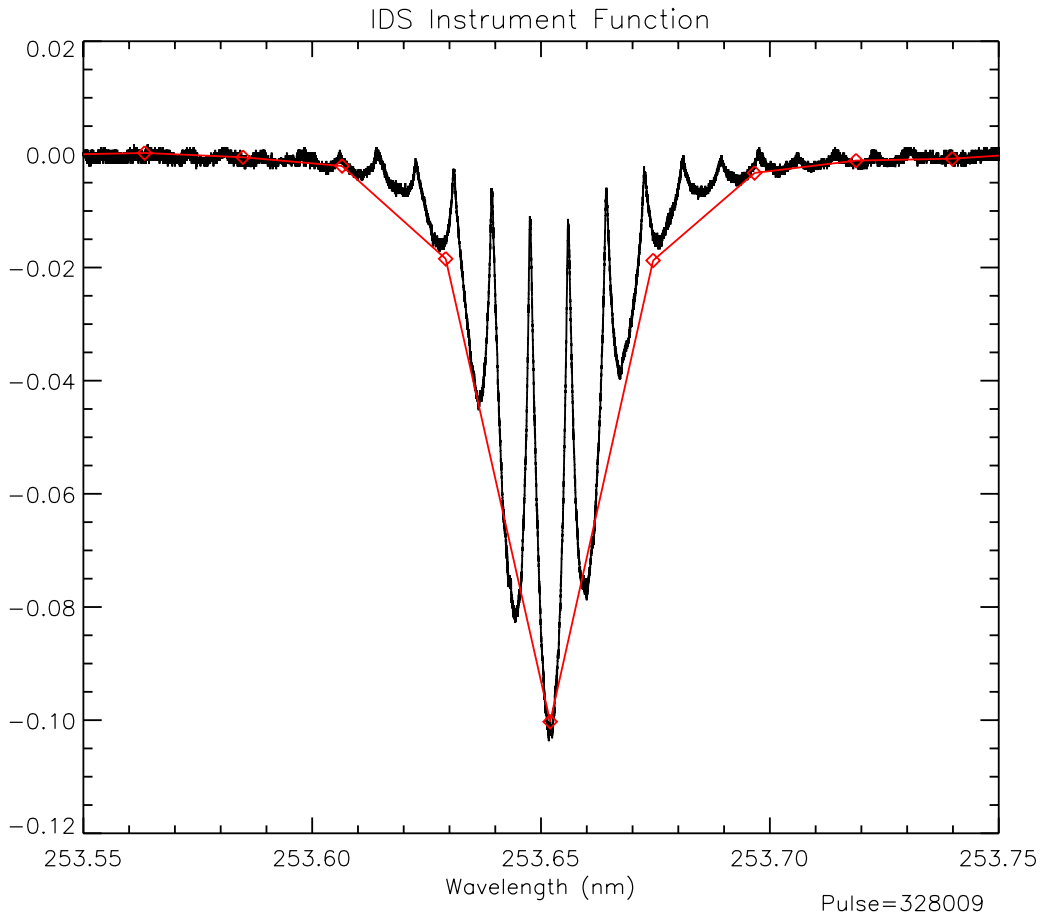

Figure 5.18: The Hg line at 253.652 nm is swept across the PMTs to measure the calibration factors. The instrument function measured with PMT 8 (black line) has a better resolution than that measured with all of the PMTs (red line). The spectral intensity measure by each PMT should be used to characterize the instrument function.

Data Analysis

The emissivity, velocity, and temperature are found from spectral intensities. The measured spectral intensities have a time varying noise caused when dV/dt is large. The noise was removed with the process described in Sec. 5.1.3. The frequency cutoff was set using the integration time of the cables, 180 ns. The plasma parameters are found by fitting the spectral intensities to Gaussians with a Marquardt method [3]. The IDL routine *CURVEFIT* was not able to converge when there was a low signal nor when there were two peaks. A

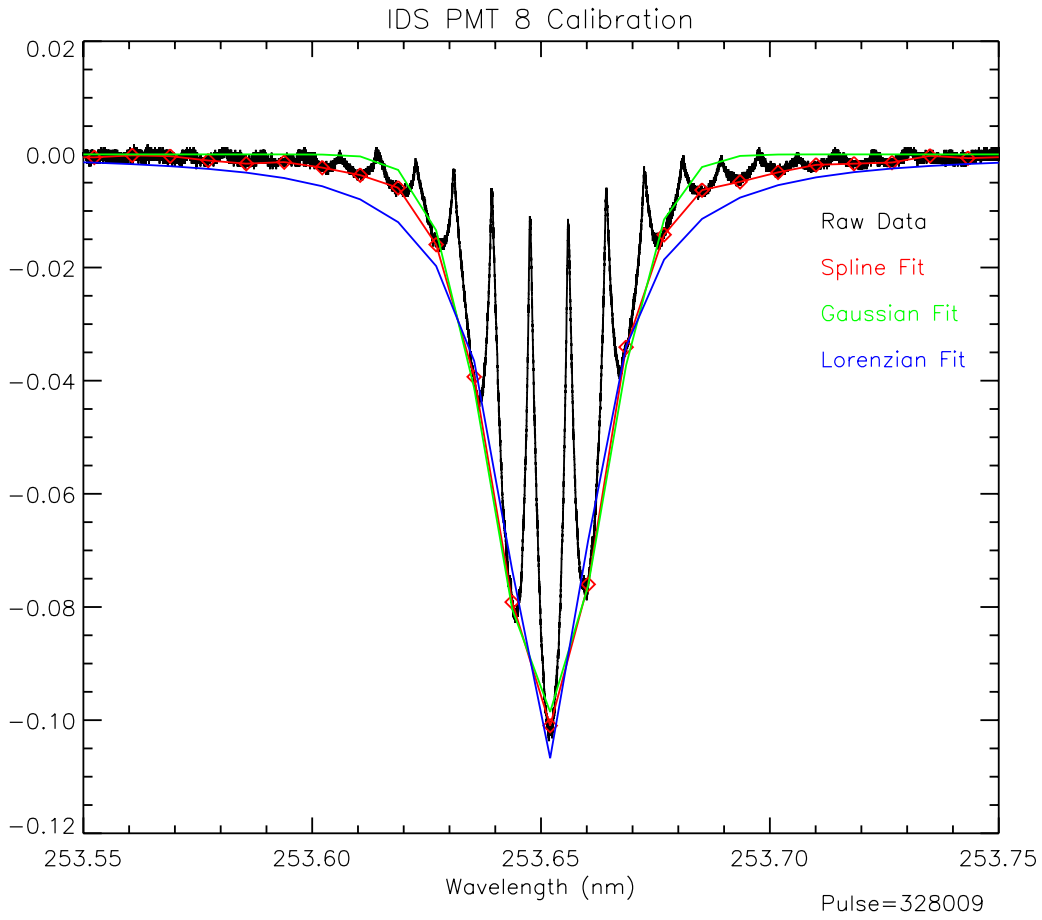

Figure 5.19: The shape of the instrument function is found by fitting different functions to it. The black line is the raw data. The colored lines are from a spline fit (red), fit to a Gaussian (green) and fit to a Lorentzian (blue). The instrument function is more like a Gaussian then a Lorentzian. The calibration factors are found with the spline fit.

time of interest was defined during which the pinch was forming and before Tungsten lines were seen by the IDS instrument. During this interval, only times where the minimum signal was less then -5 mV were curve fits attempted. The Gaussian is given by

$$y = A_0 \exp\left[-0.5\left(\frac{\lambda - A_1}{A_2}\right)^2\right] + A_3 \tag{5.53}$$

where A_0 is the amplitude, A_1 is proportional to the velocity, A_2 is the FWHM, and A_3 is proportional to the background radiation. When the instrument function are Gaussian, the

convolved spectra is a Gaussian with the A_2 given by

$$A_2^2 = FWHM_{\text{Doppler}}^2 + FWHM_{\text{Instrument}}^2 \tag{5.54}$$

where $FWHM_{\text{Doppler}}$ is the FWHM from doppler broadening and $FWHM_{\text{Instrument}}$ is the FWHM of the instrument function. The plasma parameters are given by

$$\begin{aligned} Amp &= A_0 && (5.55)\\ v_i &= \frac{c}{\cos 35^\circ}\frac{A_1 - \lambda_0}{\lambda_0} && (5.56)\\ T_i &= \frac{m_i c^2}{k\lambda_0^2}\left(A_2^2 - FWHM_{\text{Instrument}}^2\right) && (5.57) \end{aligned}$$

where Amp is the amplitude of the Gaussian. The errors in the plasma parameters, σ_{Amp}, σ_{v_i}, and $\sigma_{T\ i}$, are then given by

$$\begin{aligned} \sigma_{Amp} &= \sigma_{A\_0} && (5.58)\\ \sigma_{v_i} &= \frac{c}{\cos 35\lambda_0}\sqrt{\sigma_{A\_1}^2 + \sigma_{\lambda\_inst}^2 + \sigma_{\lambda\_Dial}^2} && (5.59)\\ \sigma_{T\_i} &= \frac{m_i c^2}{k\lambda_0^2}\sqrt{\left(2A_2\sigma_{A\_2}\right)^2 + \left(2FWHM_{\text{Instrument}}\sigma_{\text{Instrument}}\right)^2} && (5.60) \end{aligned}$$

where σ_{A_i} are the errors returned from the curve fits, σ_{λ_inst} is the error in the wavelength of each PMT, σ_{λ_Dial} is the error in λ_{dial} and $\sigma_{\text{Instrument}}$ is the error in the instrument FWHM. The amplitude, temperature, velocity and errors are calculated with an IDL routine, *ids_temp_fin3.pro* Sec. C.4.3, and written to the tree. Since the present diagnostics are not able to measure the position of the Z-pinch with a high accuracy, only chord integrated parameters are calculated for the IDS spectrometer.

5.7.4 ICCD spectrometer

The ICCD spectrometer measures the spectral intensities from 20 parallel chords through the Z-pinch. The spectrometer is versatile; three gratings may be used, two fibers are attached to different input slits, and the spectrometer has a wide wavelength range. Calibrations must be done for all the possible combinations. The spectral intensities taken with the ICCD camera are reduced with an IDL code and written to the tree. An IDL code then determines which ions are present, calculates the chord integrated plasma parameters and writes them to the tree.

Hardware

The ICCD spectrometer uses two telecentric telescopes to collect light from the plasma [25]. The telescopes are located on the 90° and 35° ports on the bottom of the vacuum chamber. Each telescope is focused at the center of the outer electrode at $z = 0$ cm. The light from parallel chords through the plasma is focused onto fiber bundles. The magnification of the telescopes is 0.5 and the étendue is 3×10^{-6} str cm^2. Two sets of fiber bundles are used. The first set consists of two identical bundles. Each bundle has 20 fused silica fibers spaced 0.62 mm apart. One bundles is mounted on the 35° and the other is one the 90° telescope. Each one uses a separate entrance slit so that temperature or velocity measures can be made. The second set, which consists of two different fiber bundles, is presently mounted on the ICCD spectrometer. One bundle has 20 fused silica fibers spaced 0.89 mm apart and the other bundle,the "Y" fiber bundle, has two groupings of ten fibers space 1.88 mm apart. Each grouping can be placed on different telescopes for absolute velocity measurements or stacked on top of each other for Zeeman splitting measurements. Filters can be used at the fiber bundle face to stop second order lines from contaminating the spectra above 400 nm or to measure different polarizations of light.

The spectrometer is a SpectraPro-500i made by Acton Research Corporation. This is a 0.5 m, Czerny-Turner spectrometer. A swing out mirror is used to select which entrance slit is used. Three gratings, 150, 240, and 3600 grooves/mm, are mounted on a turret in the spectrometer. A PI-MAX intensified CCD camera made by Princeton Instruments is used to record the spectra. A ST-133 controller with a programmable timing generator (PTG) controls the camera. Typically the exposure time is 100 ns to 1 μs. The system is operated with a PC running WinSpec which is fiber optically coupled to the spectrometer and controller.

Calibration

The ICCD spectrometer calibration is the most involved. The calibration factors for 20 chords collected with two different fibers, resolved with 3 gratings over the entire wavelength range must be found. Only the Cd lamp can be used to calibrate the ICCD spectrometer

Table 5.1: Shown are the wave regions for the ICCD spectrometer calibration factors. The first column is the Cd line used to find the calibration factors for the given wavelength region.

Cd line (nm)	Start λ (nm)	End λ (nm)
228.8022		275
326.1055	275	340
361.0508	340	420
479.9912	420	520
576.215	520	600
643.847	600	675
686.4066	675	

because none of the other lamps have a large enough region of uniform brightness. Each fiber is positioned such that all of the chords view the uniform brightness region. At least two spectra, over a 80 μs time interval at the peak of the brightness, are taken for each wavelength region with each grating. The wavelength regions used for the calibration are shown in Table 5.1 This process results in hundreds of images from which the calibration factors are found.

The vertical input slit maps to a curved line on the ICCD image. Removing this curve before each fiber is binned reduces the FWHM of the instrument function for each fiber. The total intensity of each row is used to find the bins which are used for each fiber. Regions of local maxima are the vertical location of each fiber. The location of the local minima is found. Since the light from two adjacent fibers overlap near the local minima two to three rows about the minima are not included in the bins. The curvature and bins are averaged over the acquired calibration spectra. The curvature correction is then applied and the rows are binned. The area of each spectral intensity, $A_{fiber\ i}$, is then found. The relative intensity for each fiber, rel_i is then

$$rel_i = \frac{A_{fiber\ 10}}{A_{fiber\ i}} \tag{5.61}$$

which relates all of the emissivities to fiber 10. The instrument function for each fiber is the spectral intensity of the fiber normalized by the area. This process ensures that the instrument function's centroid is at the ideal wavelength and the instrument function has a unit area. The instrument functions are asymmetric and chord dependant. The relative intensities and instrument functions are again averaged over the acquired spectra. Every setting requires a curvature correction, horizontal bin, relative intensity, and instrument function. The IDL codes, *iccd_18sep01_cal_600.pro* Sec. C.4.1 and *save_cal_18sep01_param.pro* Sec. C.4.1, are two example codes used to find the calibration values since sets must be found every time the spectrometer is calibrated. When a calibration set is not consistent with the sets which bracket it, the parameters used by the code are changed and the set is recalculate. The final calibration sets are stored in an IDL data file in the spectroscopy directory.

The calculation of the optical depth of the plasma and proper error analysis requires that the number of collected photons is known. To accomplish this, a conversion from ICCD counts to photons, β_{ICCD}, must be found.

$$I(\mathrm{counts}) = \beta_{\mathrm{ICCD}} I(\mathrm{photons}) \tag{5.62}$$

where I is the intensity. A calibrated standard was not avaliable to find this conversion factor. Instead, β_{ICCD} is found using the fact that the uncertainty in the number of measured photons, σ_I, is

$$\sigma_I(\mathrm{photons}) = \sqrt{I(\mathrm{photons})} \tag{5.63}$$

$$= \frac{\sigma_I(\mathrm{counts})}{\beta_{\mathrm{ICCD}}} \tag{5.64}$$

where I is the measured brightness. After some algebra β_{ICCD} becomes

$$\beta_{\mathrm{ICCD}} = \frac{\sigma_I^2(\mathrm{counts})}{I(\mathrm{counts})} \tag{5.65}$$

One method of finding β_{ICCD} is to use a region of the measured spectral intensity from each pulse where there is no line emission to measure and I (counts) and σ_I(counts) [18].

The calibration factor was verified by taking a large number of spectra of a constant light source (a bright LED) and measuring I (counts) and σ_I (counts). Since the ICCD

image consists of a background and the desired spectra, a large number of background (LED off) images are also taken to reduce error of the subtracted spectra. Various ICCD spectrometer configurations were tested. Fifty spectra are taken for each configuration, twenty background and thirty with the LED off. The average background image is

$$B_{ave} = \frac{1}{N} \sum_{i=1,N} B_i \tag{5.66}$$

where B_{ave} is the average background, 512x512 array, for N images. The uncertainty of the background σ_B is

$$\sigma_B^2 = \frac{1}{N-1} \sum_{i=1,N} (B_i - B_{ave})^2 \tag{5.67}$$

Once the background is found, the spectra, C from the LED is found.

$$C_i = A_i - B_{ave} \tag{5.68}$$

An average spectra C_{ave} is found using Eq. 5.66. The uncertainty in C_{ave}, σ_C, is found using Eq. 5.67. The uncertainty from the background subtraction is removed with

$$\left(\sigma_C^{\text{corrected}}\right)^2 = \sigma_C^2 - \sigma_B^2 \tag{5.69}$$

where $\sigma_C^{\text{corrected}}$ is the uncertainty from photon statistics. The intensity of each wavelength found by summing across all fibers.

$$I_j = \sum_i (C_{ave})_{ij} \tag{5.70}$$

where j corresponds to the column (wavelength) and i corresponds to the row. The uncertainty in I due to photon statistics becomes

$$\sigma_I^2 = \sum_i \left(\sigma_C^{\text{corrected}}\right)_{ij}^2 \tag{5.71}$$

β_{ICCD} is found using Eq. 5.65. The result of a pseudo-calibration are shown in Fig. 5.20. The value of measured β_{ICCD} agrees with the estimated value.

Image Reduction

The images taken by the ICCD spectrometer are converted into the spectral intensities of the 20 chords. An image taken with the ICCD is shown in Fig. 5.21 An IDL code is used

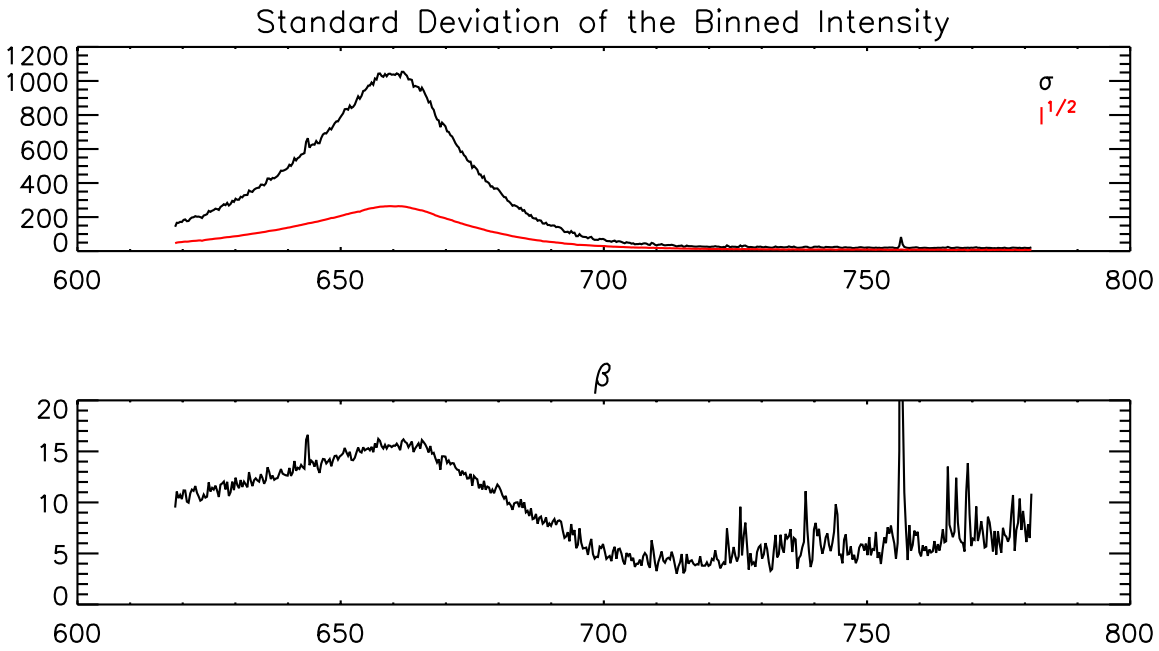

Figure 5.20: β_{ICCD} is found with the standard deviation and square root of the intensity of an led. The calculated value of β agrees with the method described in [18].

apply the calibration factors. The images are read by IDL with *READ_PRINCETON*, a routine written at the University of Chicago, which reads the image and wavelength from the WinSpec file. The curvature is removed from the image with the appropriate correction and the rows are binned according to the chords. The binned data, relative intensity, instrument function and calibration data are then written to the tree with IDL routines, *w_iccd_data_pro.pro* Sec. C.4.3, *iccd_reduce.pro* Sec. C.4.3 *repair_iccd_node.pro* Sec. C.4.3, and *iccd_save.pro* Sec. C.4.3.

Surveys have identified impurities which can be used for passive spectroscopy. The line radiation has been identified by comparing the relative intensity of multiple lines to the NIST database. The lines used for passive spectroscopy are from wavelength regions where other ions do not contaminate the line radiation. The level of the radiation, shown in Fig. 5.22, is below the black body radiation level, shown in Fig. 5.14. This shows that the plasma is optically thin which simplifies the analysis of the spectral intensities. The

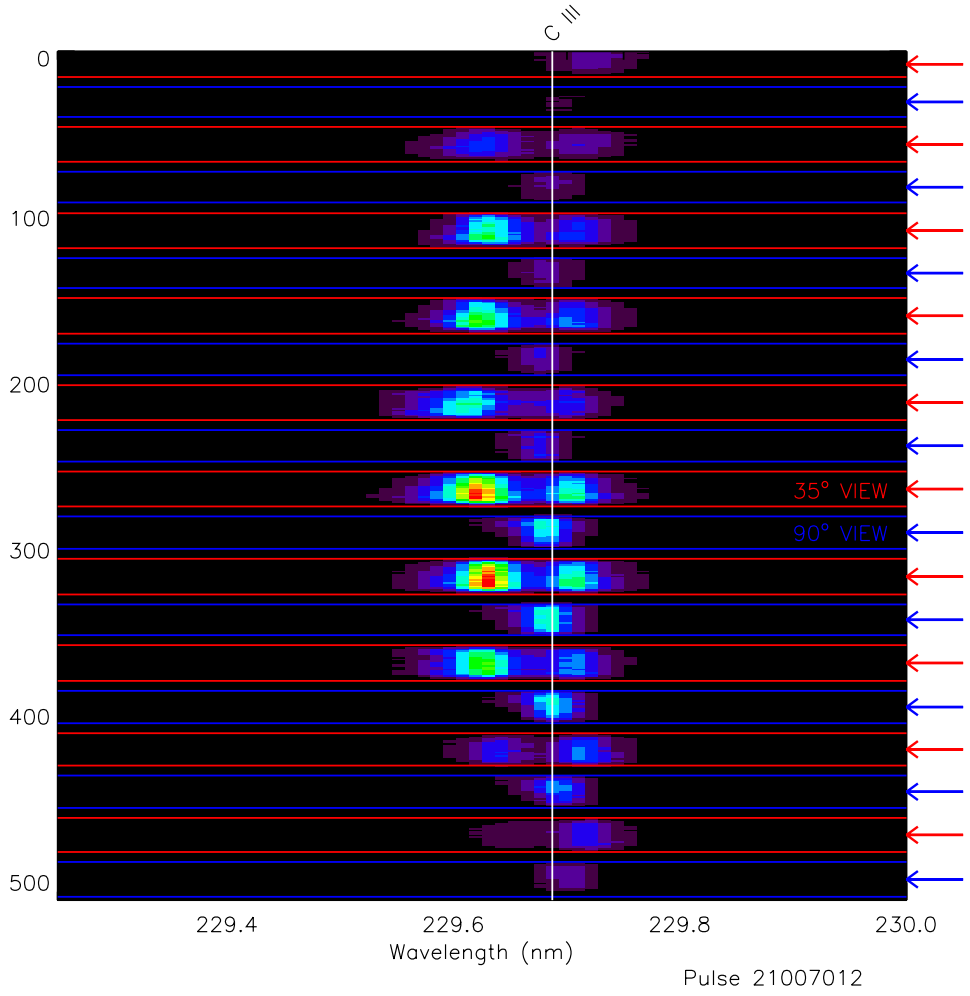

Figure 5.21: The image of the spectral intensities taken with the split fibers. The curvature has been removed from this image. The bins are located between the horizontal lines.

line radiation also decreases with time. The gain of the ICCD spectrometer is adjusted to maximize the number of counts.

Chord Integrated Parameters

The chord integrated plasma parameters are found using the measured spectral intensities. Many different ions are used to measure the plasma parameters. Shown in Table 5.2 is a list of the ions which are used to measure the plasma parameters. Since there are a large number

Table 5.2: The ions which are used to find plasma parameters. The lines which are wide have a w following the relative intensity. Lines which differ only by J have a * following the relative intensity.

Ion	λ (nm)	Relative Intensity	Ion	λ (nm)	Relative Intensity
B III	206.5776	550	O V	559.789	130
	206.7233	450	O V	278.101	1000
C III	229.687	800		278.699	920
C III	464.7418	600		278.985	775
	465.0246	520	O VI	381.135	
	465.1473	375	O VI	383.424	
C IV	465.83	200w	O VII	255.940	
C IV	580.131	250	N III	268.218	250w
	581.197	200	N III	409.736	250
C V	227.089	40		410.339	200
	227.727	5	N IV	638.075	150
	227.792	20	N IV	264.562	250
C VI	343.369			264.619	300*
C VI	529.1			264.619	300*
He II	468.54	4		264.698	350*
	468.57	30		264.698	350*
O III	326.0857	200		264.698	350*
	326.5329	300	N IV	347.872	570
	326.7204	80		348.300	500
O III	375.4696	150		348.493	400
	375.7232	80	N V	298.078	150w
	375.9875	250		298.131	250w
O IV	306.343	460		299.843	60w
	307.160	410	N VI	205.61	
	.		N VI	289.64	

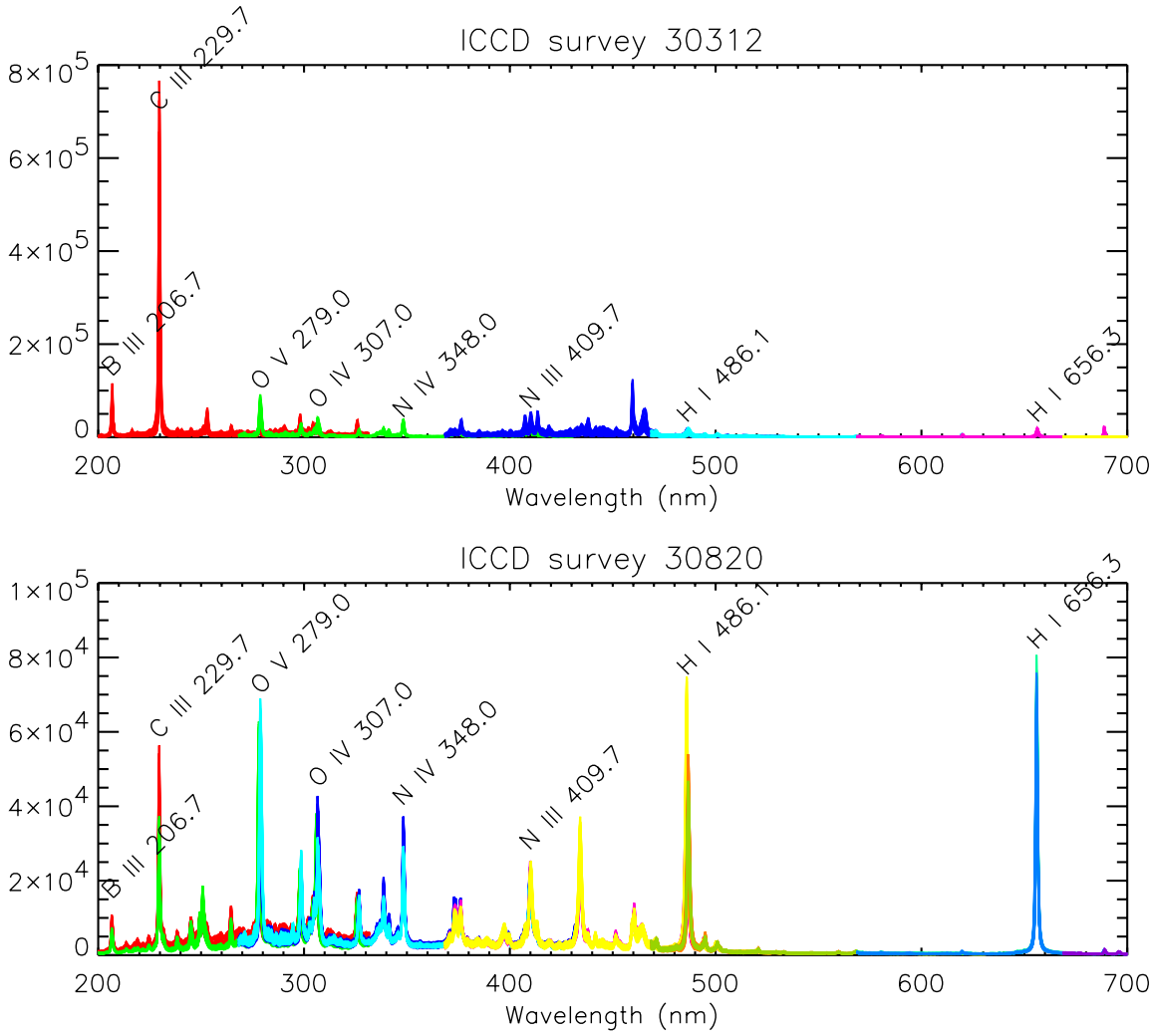

Figure 5.22: The spectra of the Z-pinch are measured with the ICCD spectrometer. The series taken on March 12, 2003 (upper) uses a 10 μs gate. The series taken on August 20, 2003 (lower) used a 20 μs gate. The intensity of the line radiation has decreased.

of different ions which are used an IDL routine, *find_iccd_raw_param_SYIAH.pro* Sec. C.4.3, is used to determine which lines are present on a given plasma pulse. This routine looks for spectral intensities near each unshifted wavelength which are greater than 10 times the background noise level of the spectra.

When this condition is met the plasma parameters are found by fitting an instrument

widened Gaussian to the spectral intensity. The instrument functions on the ICCD spectrometer are asymmetric. The spectral intensities of single lines are fit to

$$y = \int_{-\triangle\lambda}^{\triangle\lambda} \left\{ A_0 \exp\left[-0.5 \left(\frac{\lambda - \lambda' - A_1}{A_2} \right)^2 \right] + A_3 \right\} I_{\text{instrument}}(\lambda') d\lambda' \tag{5.72}$$

where the A_i's are related to the plasma parameters, with a Marquardt method [3]. When the line is a doublet, an additional Gaussian is added inside the brackets of Eq. 5.72.

$$\begin{aligned} y = \int_{-\triangle\lambda}^{\triangle\lambda} \Bigg\{ & A_0 \exp\left[-0.5 \left(\frac{\lambda - \lambda' - A_1}{A_2} \right)^2 \right] \\ & + RI_{\text{NIST}} A_0 \exp\left[-0.5 \left(\frac{\lambda - \lambda' - A_1 + \triangle\lambda_{\text{NIST}}}{A_2} \right)^2 \right] + A_3 \Bigg\} I_{\text{instrument}}(\lambda') d\lambda' \end{aligned} \tag{5.73}$$

where RI_{NIST} is the ratio of the relative intensities and $\triangle\lambda_{\text{NIST}}$ is the difference between the central wavelengths from the NIST data base. A third Gaussian is added for triplets. Since the same fit parameters are used for singlets, doublets and triplets, the plasma parameters are found with

$$E = \sqrt{2\pi} A_0 \tag{5.74}$$

$$\mathbf{v_i} = \frac{c}{\cos\theta} \frac{A_1 - \lambda_0}{\lambda_0} \tag{5.75}$$

$$T_i = \frac{m_i c^2 A_2^2}{k \lambda_0^2} \tag{5.76}$$

$$Br = A_3. \tag{5.77}$$

This method has the additional benefit that the fits are poor when other lines contaminate the spectra. *find_iccd_raw_param_SYIAH.pro* uses IDL codes like *iccd_raw_multi_emis.pro* Sec. C.4.3 to calculate the plasma parameters and write them to the tree.

5.7.5 Deconvolution Technique

The calculation of the local plasma parameters involves deconvolving the measured spectra [20]. Since the process described in Sec. 5.7.1 cannot be simply reversed, the following procedure is used to find the local plasma properties. This section presents the relevant assumptions for the model. The deconvolution technique and error analysis are then described. The last section compares the results of the method to known profiles.

Shell Model

A shell model is assumed to represent the plasma. The plasma is partitioned into discrete shells where the local amplitude of the emissivity A_j, ion temperature T_j, and velocity $\mathbf{v}_j$ are constant in each shell [2]. The spectral intensity from shell j along the line of sight of chord i is then described by

$$e_{ij}(\lambda) = \frac{A_j}{\sqrt{2\pi} w_j} \exp\left[-\frac{(\lambda - \lambda_0 - |\mathbf{s}_i \cdot \mathbf{v}_j| \lambda_0 / c)^2}{2\ w_j^2} \right] + B_j \ , \tag{5.78}$$

where λ_0 is the unshifted wavelength of the emitted line radiation, $\mathbf{s}_i$ is a unit vector in the direction of the line of sight, w_j is proportional to the FWHM of the spectral intensity, and B_j is an offset of the spectra due to broadband radiation. The temperature is related to w_j by

$$w_j^2 = \frac{k T_j \lambda_0^2}{m_i c^2} \ , \tag{5.79}$$

where k is Boltzmann's constant and m_i is the ion mass. The shells are assumed to be concentric circles, where the axis of the plasma coincides with the center of the shells. The shell geometry for the 35° telescope is shown in Fig. 5.23. The outer radius of shell j is given by $r_j = x_{\text{chord}}(j) + \delta \triangle r$ where $x_{\text{chord}}(j)$ is the impact parameter of chord j, $\triangle r$ is the spacing between chords and shells, and δ is the relative position of the viewing chord in each shell. δ is set to 1 for the deconvolved profiles from the ICCD spectrometer. The outer radius of shell 1 is set by the extrapolated zero crossing of the measured emissivity, usually $r_1 = r_2 + 2\triangle r$. The radiation measured by each chord is the sum of the contributions from each shell that the chord intersects, shown in Fig. 5.23, when the plasma is optically thin. The collected spectral intensity for chord i is given by

$$E_i(\lambda) = \sum_j e_{ij}(\lambda) L_{ij} \ , \tag{5.80}$$

where L_{ij} is the length of chord i through shell j. The collected spectral intensity is widened by the instrument function.

$$M_i(\lambda) = \int_{-\triangle\lambda}^{\triangle\lambda} E_i(\lambda - \lambda') F_i(\lambda') d\lambda' \ , \tag{5.81}$$

where M_i is the instrument widened spectral intensity, F_i is the instrument function of chord i measured during the calibration of the ICCD spectrometer and $2\triangle\lambda$ is the span of the

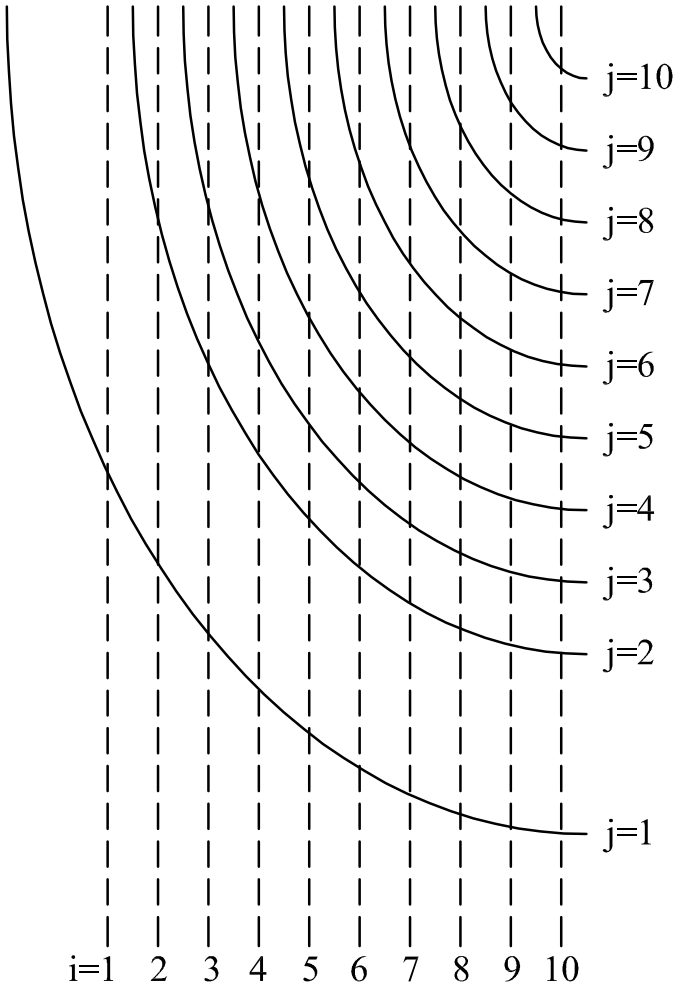

Figure 5.23: The 35° view of a quarter of the plasma showing the chord and shell locations when the plasma centered in the machine. The dashed lines are the sight lines of the 35° viewing telescope. The ovals are the outer edges of each shell when viewed from the 35° viewing telescope.

measured instrument function. Coma in the spectrometer causes the instrument function to be asymmetric as shown in Fig. 5.24. The FWHM of the instrument function, measured with the Cd I line at 228.8 nm, varies from 0.030 nm for the center chords to 0.047 nm for the edge chord. The inclusion of the instrument function in the analysis causes the spectral intensities to be a function of the impact parameter and the viewing chord even when the angle between the velocity vector and the lines of sight of the chords does not change.

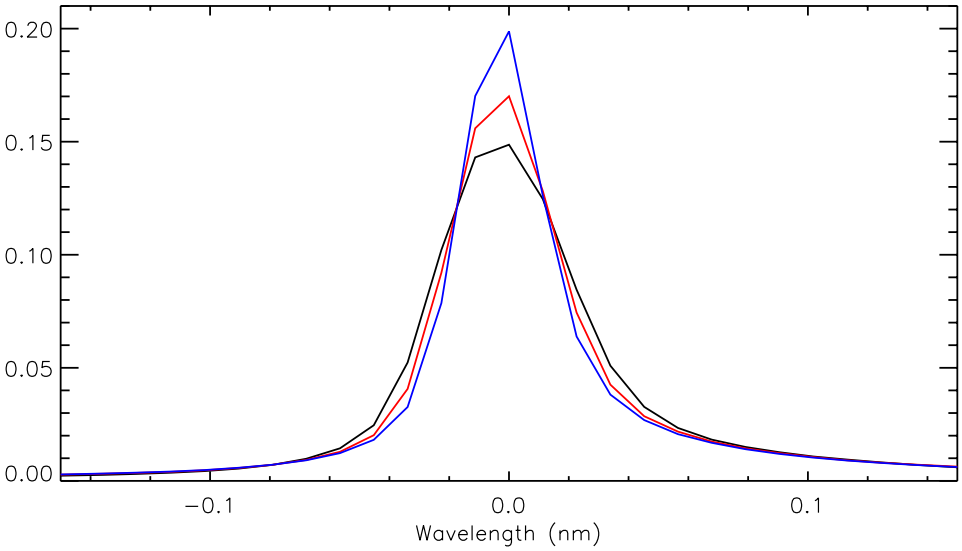

Figure 5.24: The instrument function varies across the chords. Shown are the measured instrument functions of chord 1 (black), chord 5 (red) and chord 10 (blue) using the Cd I line at 228.8 nm. The width of the instrument function decreases for the central chords.

Deconvolution Procedure

The plasma parameter profiles cannot be accurately found by simply reversing the steps of the previous section because of difficulties with the removal of the instrument function. Different methods of deconvolving the measured spectra have been tried [4, 48]. The methods are able to deconvolve test spectral intensities as shown in Fig. 5.25. When noise consistent with the experimental levels are added, the methods are not able to deconvolve the spectral intensities as shown in Fig. 5.26. None of the methods have been able to remove the instrument function from noisy data.

This difficulty precludes the use of Abel or matrix methods because the instrument function must be removed before using these methods. Instead an instrument widened Gaussian is defined for each shell

$$m_{ij}(\lambda) = \int_{-\triangle\lambda}^{\triangle\lambda} e_{ij}(\lambda - \lambda')F_i(\lambda')d\lambda' \ . \tag{5.82}$$

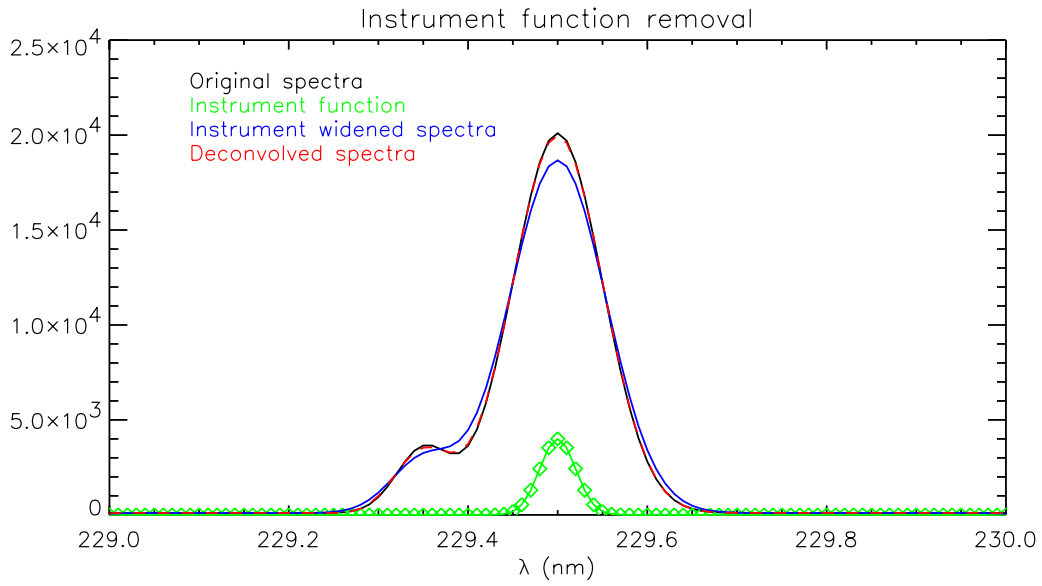

Figure 5.25: Shown are the results of removing the instrument function from simulated spectral intensities. The original spectra (black) was widened with the instrument function (green). The instrument function was then removed from the instrument widened spectra (blue). The resulting spectra (red) from this process matched the original spectra.

An equation similar to Eq. (5.81) can be written as

$$M_i(\lambda) = \sum_j m_{ij}(\lambda) L_{ij} \ . \tag{5.83}$$

Since the length matrix L_{ij} is triangular, back substitution can be used to find the plasma parameters.

By viewing the plasma with telecentric telescopes, only the horizontal location of the plasma affects the impact parameter of each viewing chord. The location is found by iteratively deconvolving the data. Guesses for the center and the edge of the pinch are made, which sets the geometry. The length matrix L_{ij} is calculated and the plasma parameter profiles for this geometry are then found.

The edge chord measures only the emission from the outermost shell. Let $i = 1$ be the outermost chord and $j = 1$ the outermost shell. All of the terms L_{1j} are zero except for

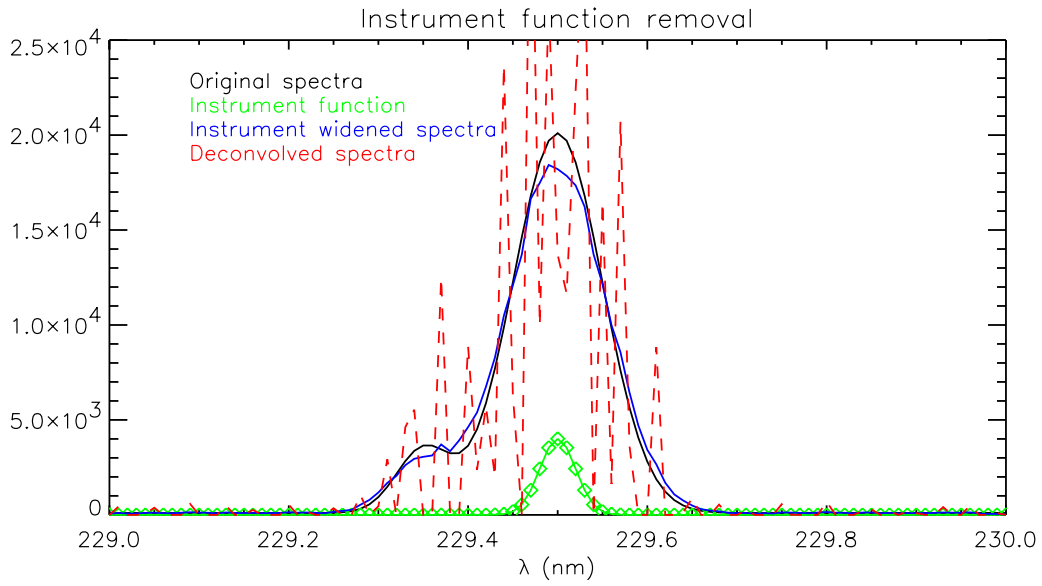

Figure 5.26: Shown are the results of removing the instrument function from the spectral intensities with noise. The original spectra (black) was widened with the instrument function (green). Noise consistent with Poisson statistics was then added to the instrument widened spectra (blue). The instrument function was then removed from the instrument widened spectra. The resulting spectra (red) from this process does not match the original spectra.

$j = 1$. The measured spectral intensity of the outer chord is given by

$$\begin{aligned} M_1(\lambda) &= L_{11} \int_{-\triangle\lambda}^{\triangle\lambda} e_{11}(\lambda - \lambda') F_1(\lambda') d\lambda' \\ &= L_{11} m_{11}(\lambda) \ . \end{aligned} \tag{5.84}$$

The plasma parameters for the outer shell are found with a least squares fit of an instrument widened Gaussian to $M_1(\lambda)/L_{11}$ using a Marquardt method with equal weights given to each point. The wings of the spectral intensity are given a greater weight when Poisson statistics are used as weights in the fits, shown in Fig. 5.27. The physics which are being studied affect the center of the line shape more than the wings. The fits are done with equal weights for all values to capture the Gaussian's parameters. Poisson statistics should be used for line broadening which is Lorentzian in character.

The plasma parameters of the inner shells are determined by removing the contribution from the outer shells and fitting an instrument widened Gaussian to the remaining data. Let

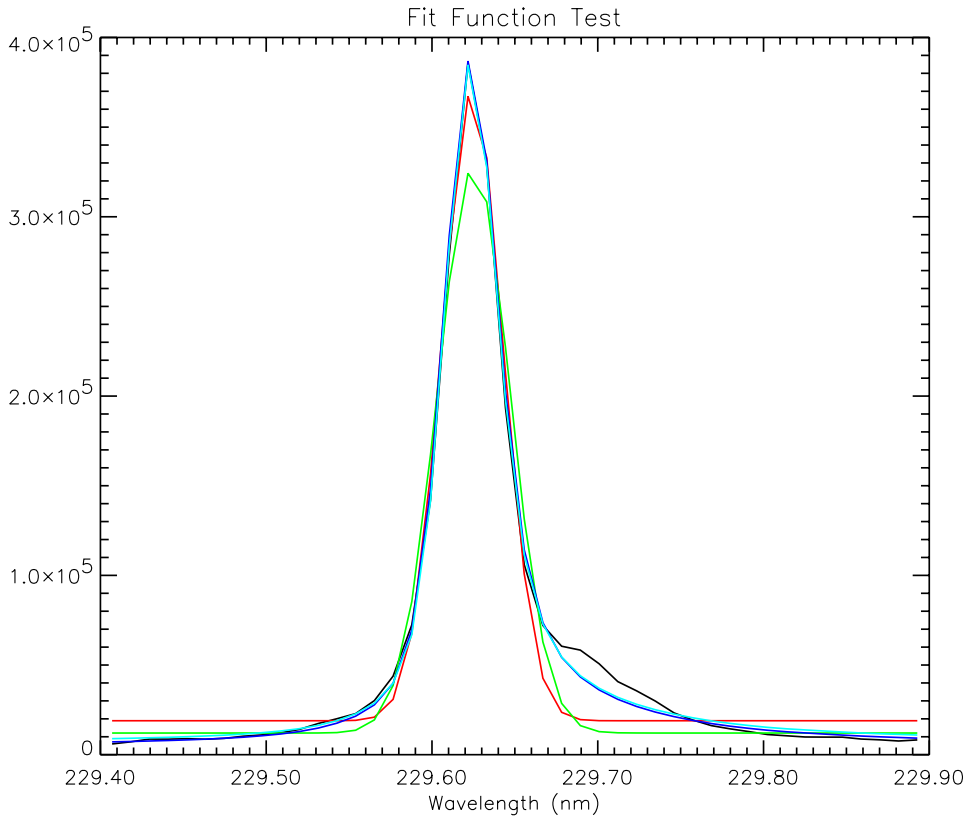

Figure 5.27: The weights used in the fitting routines affect the fitted spectra. The black curve is the measured spectra from pulse 726025 chord 1. The fitted spectra from Eq. 5.90 using constant weights (red) and Poisson statistics (green) show the largest intensities are not being modeled with Poisson statistics. The effect is hard to see when Eq. 5.90 is used with (blue) and without (cyan) Poisson statistics.

chord i be a chord in the plasma and shell j be the corresponding shell in the plasma. An instrument widened Gaussian for the outer shell(s), 1 to $j-1$, contribution to the measured spectral intensity M_i^{outer} is found with

$$M_i^{\text{outer}}(\lambda) = \sum_{j'<j} m_{ij'}(\lambda) L_{ij'} \ , \tag{5.85}$$

where $m_{ij'}$ is calculated using the plasma parameters from the previous shells and Eq. (5.82). The emission from the shell is given by removing the outer shells' contribution

$$m_{ij}(\lambda) = \max\left(\frac{M_i(\lambda) - M_i^{\text{outer}}(\lambda)}{L_{ij}}, 0\right) \ . \tag{5.86}$$

The plasma parameters for shell j are found with a Marquardt method. This procedure is repeated for each successive chord until the shell axis is crossed. Once the axis is crossed the plasma parameters for the inner two shells are also found using the spectral intensities measured by the next two chords. The deconvolution procedure is also repeated starting from the other side, beginning at chord 20 and decreasing the chord index.

The two deconvolution procedures yield two profiles of the emissivity, ion temperature, and ion velocity for the assumed geometry. The center of the plasma is found by adjusting the guess of the pinch center until the emissivity and the velocity of the inner two shells from the left and right deconvolutions converge.

Error Analysis

There are two main sources of error with this method, noise in the measured spectral intensities and uncertainties in the calculation of the length matrix, which results from the uncertainty in the pinch geometry. The data are fit with a Marquardt method with equal weights given to each point. The actual uncertainty for each spectral intensity is proportional to the square root of the intensity. When the actual uncertainties are used as weights, the fitted spectra had a lower amplitude at the maximum measured spectral intensity and fit the wings of the spectra.

The errors in the plasma parameters are found for each source of uncertainty. The uncertainty of the measured spectra $\sigma_{Mi}(\lambda)$ is given by

$$\sigma_{Mi}^2(\lambda) = \left[\beta_i \sqrt{M_i(\lambda)}\right]^2 + \left[\sqrt{M_{Bi}}\right]^2, \tag{5.87}$$

where β_i is a multiplier for each chord found by measuring the standard deviation of the spectra where there is no line radiation [18] and M_{Bi} is the level of the background of each chord. The error in the spectral intensities of the inner shell $\sigma_{Nmij}(\lambda)$ due to noise in the measured spectral intensities is given by

$$\sigma_{Nmij}^2(\lambda) = \left[\frac{\sigma_{Mi}(\lambda)}{L_{ij}}\right]^2 + \left[\frac{\sigma_{N\text{outer}}(\lambda)}{L_{ij}}\right]^2, \tag{5.88}$$

where $\sigma_{N\text{outer}}(\lambda)$ is the uncertainty from $M_i^{\text{outer}}(\lambda)$ which is given by

$$\sigma_{N\text{outer}}^2(\lambda) = \sum_{j'<j} \left\{ \sum_l \left[\frac{\partial m_{ij'}(\lambda)}{\partial a_{j'l}} \sigma_{Nj'l}^a L_{ij'}\right]^2 \right\} \tag{5.89}$$

where $a_{j'l}$ is parameter l of shell j' and $\sigma^a_{Nj'l}$ is the error of $a_{j'l}$ due to the uncertainty in the measured spectral intensities. Once the uncertainties at each wavelength are found, the error of each parameter σ^a_{Njl} can be found by taking the inverse of the curvature matrix.

The errors in the fit parameters are found by using the inverse of the curvature matrix as described in Bevington [3]. If the parameters are not correlated, the curvature matrix is diagonally dominant and the inverse is easy to calculate. The parameters used in Eq. 5.53 are correlated, seen in Fig. 5.28. The errors found using the inverse of the curvature matrix are on the solid contours. The correct errors are found with the dashed lines on the plots. The correlation of the terms causes the major axis of the ellipse to be off of the axes. Better fitting functions are

$$y = \frac{A_0}{A_2}\exp\left[-0.5\left(\frac{\lambda - A_1}{A_2}\right)^2\right] + A_3 \quad \text{or} \tag{5.90}$$

$$y = \int_{-\triangle\lambda}^{\triangle\lambda}\left\{\frac{A_0}{A_2}\exp\left[-0.5\left(\frac{\lambda - \lambda' - A_1}{A_2}\right)^2\right] + A_3\right\} I_{\text{instrument}}(\lambda')d\lambda' \tag{5.91}$$

because the correlation between parameters is reduced, seen in Fig. 5.29. The terms are still correlated since the major axis of the ellipses is not along an axis. The correlation of more than two terms is reduced since the two sets of curves are similar. Taking the inverse of the curvature matrix correctly calculates the errors for Eq. 5.91. While the IDL routine *CURVEFIT* applies the Marquardt method correctly, the errors found for each term are incorrect when the terms are coupled. A separate routine is used to find the uncertainties in the fits to Gaussians.

The uncertainties in the length matrix σ_{Gij} are calculated using the uncertainty in the center location, maximum shell radius, and the location of the chord through each shell. The error $\sigma_{Gmij}(\lambda)$ due to the uncertainties in the geometry of the pinch is given by

$$\sigma^2_{Gmij}(\lambda) = \left[\frac{\sigma_{G\text{outer}}(\lambda)}{L_{ij}}\right]^2 + \left\{\frac{\left[M_i(\lambda) - M_i^{\text{outer}}(\lambda)\right]^2\ \sigma_{Gij}}{L^2_{ij}}\right\}^2, \tag{5.92}$$

where the uncertainty in the spectral intensity from the outer shells $\sigma_{G\text{outer}}(\lambda)$ is given by

$$\sigma^2_{G\text{outer}}(\lambda) = \sum_{j'<j}\left\{\left[m_{ij'}(\lambda)\sigma_{Gij'}\right]^2 + \sum_l\left[\frac{\partial m_{ij'}(\lambda)^2}{\partial a_{j'l}}\ \sigma^a_{Gj'l}L_{ij'}\right]^2\right\} \tag{5.93}$$

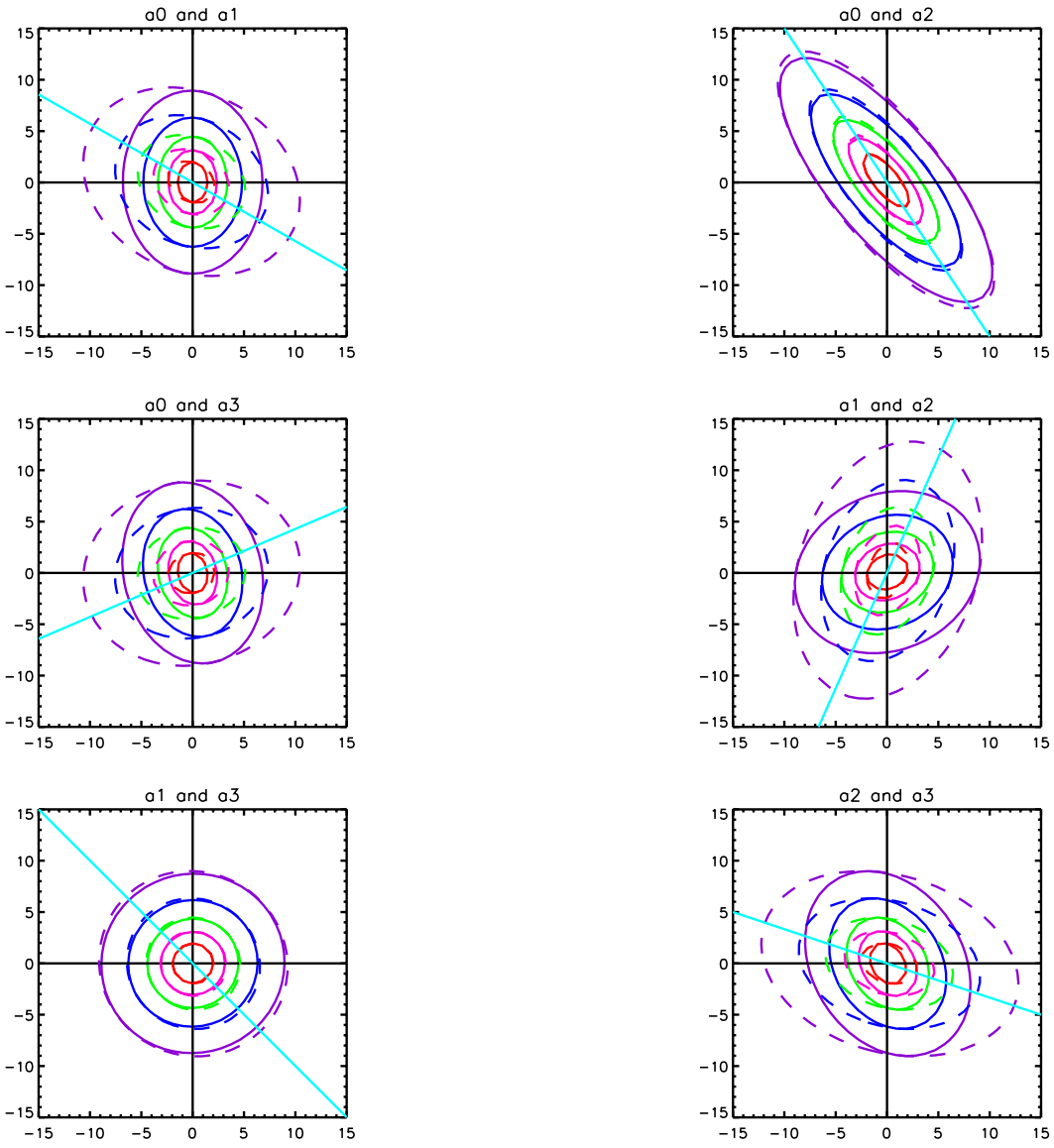

Figure 5.28: Shown is χ projected onto A_i, A_j subspaces when the amplitude is used as a fit parameter. The solid lines are contours of χ for fixed values of $A_{\neq i,j}$. The dashed lines are the minimum value of χ for each A_i and A_j ($A_{\neq i,j}$ are optimized for each A_i and A_j). The nonlinearity of the parameters is seen in the differences between the two sets of curves. The major axis is not aligned with either axis indicating that the terms are correlated. The actual error is larger than the calculated error

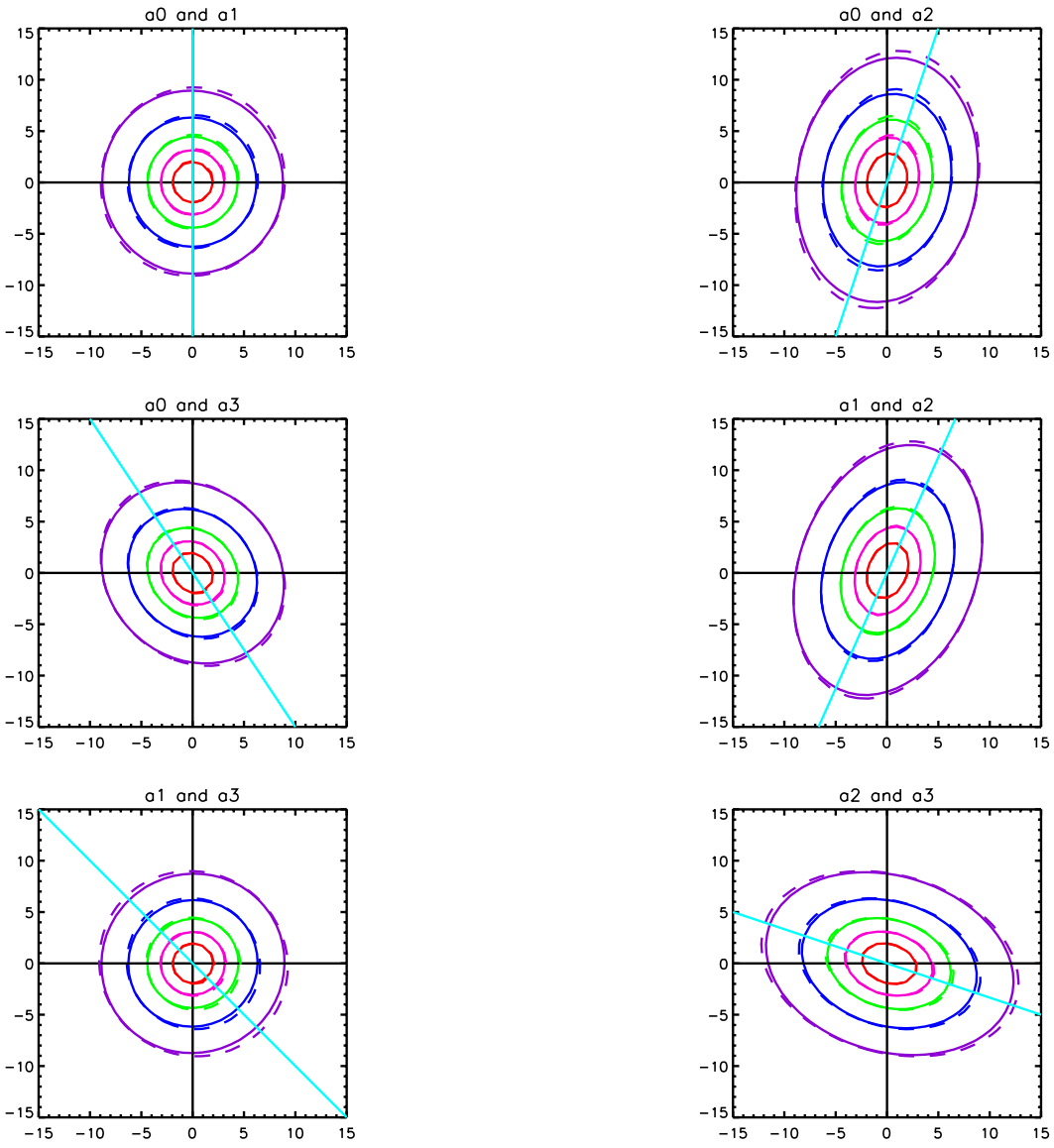

Figure 5.29: Shown is χ projected onto A_i, A_j subspaces when the area is used as a fit parameter. The solid lines are contours of χ for fixed values of $A_{\neq i,j}$. The dashed lines are the minimum value of χ for each A_i and A_j when $A_{\neq i,j}$ are optimized for each A_i and A_j. The nonlinearity of the parameters is seen in the differences between the two sets of curves. The major axis is not aligned with either axis indicating that the terms are correlated. The correct errors are calculated for this set of equations.

where $\sigma^a_{Gj'l}$ is the error of each parameter due to the uncertainty in the geometry. The error of each parameter due to the uncertainties in the geometry is found with the same method as with σ^a_{Njl}.

The total error in each parameter σ^a_{jl} is

$$\sigma^a_{jl} = \sqrt{\left(\sigma^a_{Njl}\right)^2 + \left(\sigma^a_{Gjl}\right)^2} \ . \tag{5.94}$$

The total errors in the fit parameters are then used to calculate the errors in the plasma parameters, similar to Eq. 5.60. This method of finding the errors show the influence of each uncertainty on each parameter.

Results of the Deconvolution Technique

The deconvolution process is tested with various synthetic profiles of emissivity, velocity, and temperature. The local spectral intensity along each line of sight, $e(r, \lambda)$, was found using Eq. (5.42). The chord integrated spectral intensity for each wavelength, $E(x, \lambda)$, were found with

$$E(x, \lambda) = 2 \int_x^R e(r, \lambda) \frac{r dr}{\sqrt{r^2 - x^2}} \ . \tag{5.95}$$

$E(x, \lambda)$ was widened with the instrument function using Eq. (5.81). This process of generating synthetic spectral intensities tests the effect of assuming the plasma parameters are a constant in each shell. Random noise, with a mean of zero and standard deviation of 1, is multiplied by $\sigma_{M_i}(\lambda)$, calculated using Eq.(5.87), and added to the synthetic spectral intensities. The spectra were then deconvolved using the procedure described.

The agreement between the synthetic spectra and fitted spectral intensities is shown in Fig. 5.30 The spectral intensities from the edge chord, shown in the top plot, is skewed from the instrument function. The contribution from the outer shells is shown as the green line in the bottom plot. The plasma parameters are found from a curve fit to the blue trace. The fitted spectral intensity matches the synthetic shape.

Many synthetic profiles of emissivity, velocity and temperature were used to test the technique. The errors due to the geometry were tested by setting the parameters to incorrect values of the center and edge of the pinch. Different center locations had the largest effect on

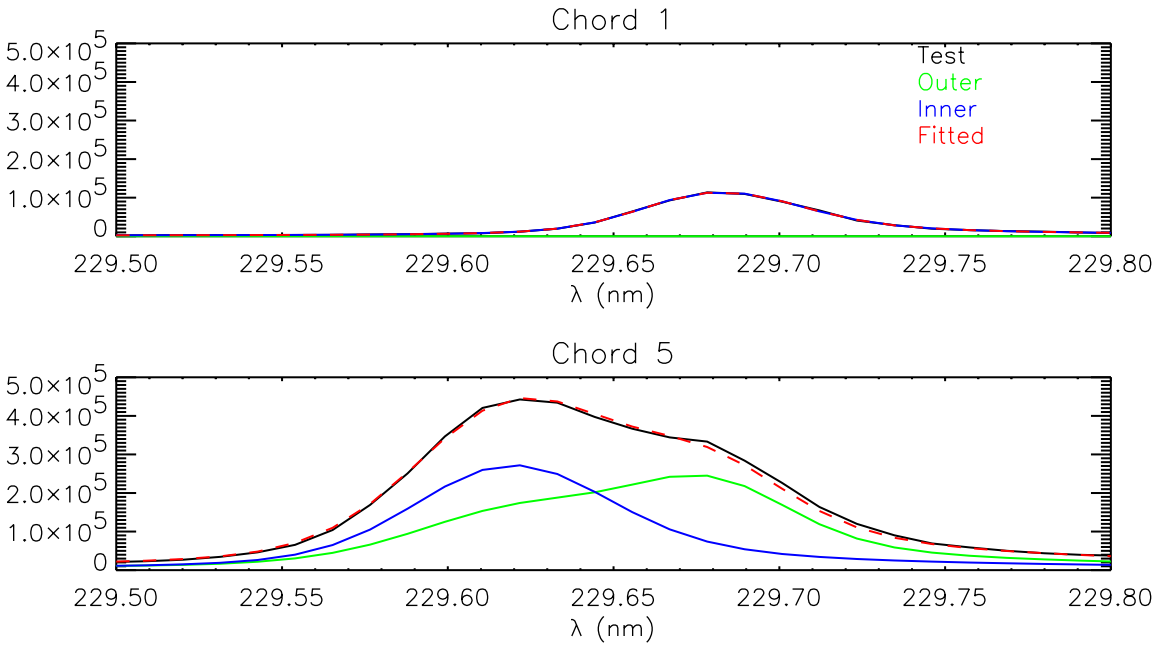

Figure 5.30: The deconvolution technique is able to recreate the spectral intensities. Shown are the synthetic spectral intensities (black) and the spectral intensities from the deconvolution technique (red). The contribution from the outer shells (green) has a complicated shape. The skewness in the spectral intensity from the inner shell (blue) is from the instrument function.

the plasma parameters in the central shells, which justifies using the inner shell parameters to set the pinch geometry. The value of the parameters for the edge shells changed by a small amount. Figure 5.31 shows a sample synthetic profile and the results from the deconvolution technique. The appropriate δ is a function of the gradients in the profiles. When there is a large gradient in the velocity, the measured temperature increases because a velocity gradient within a shell will widen the spectral intensity. The error calculation was verified by applying a large number of different noise sets to the spectral intensities calculated with this profile, applying the deconvolution technique with geometries within the uncertainties of the pinch geometry, and finding the average and standard deviation of the plasma parameters. The calculated error bars of the plasma parameters are slightly larger than the standard deviation found with the multiple deconvolutions.

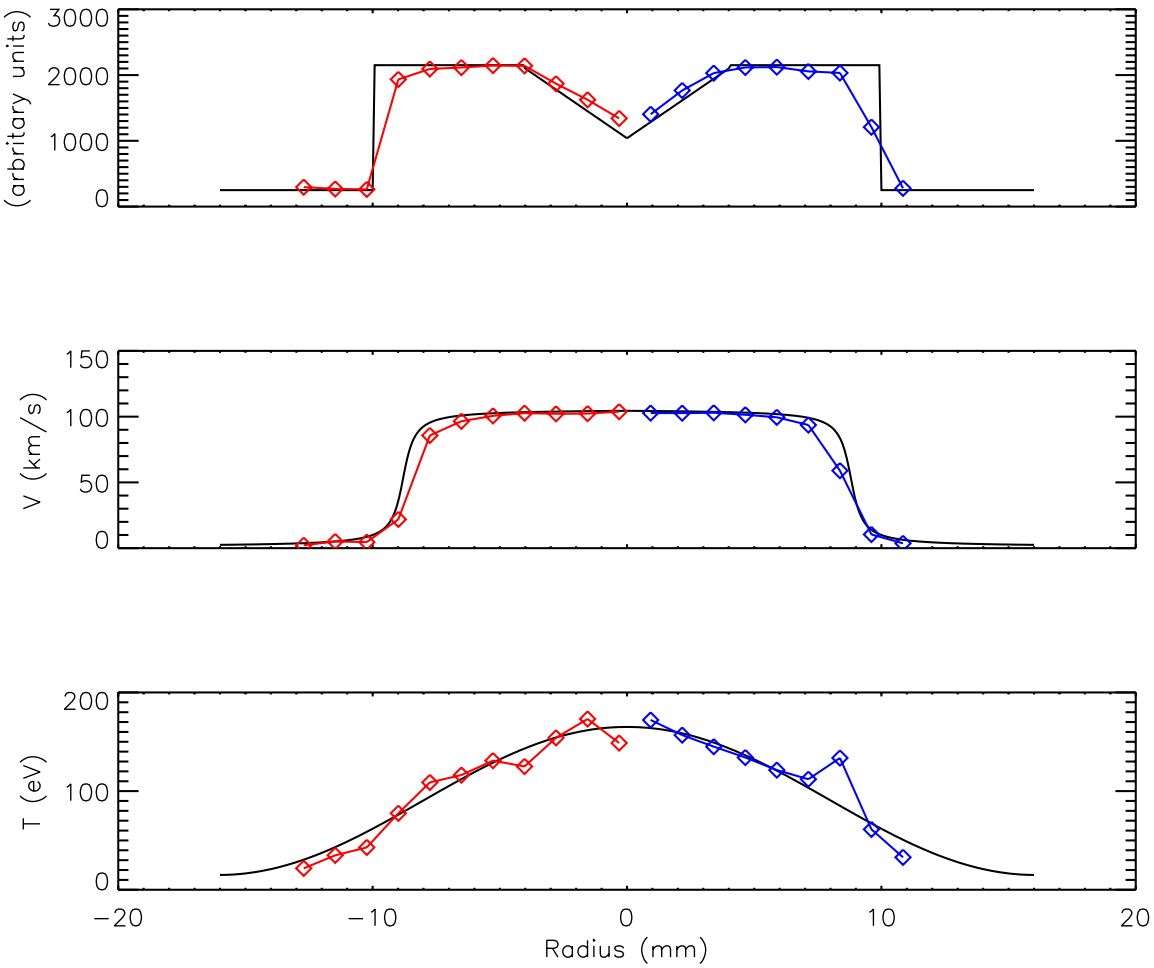

Figure 5.31: The deconvolution technique is able to reconstruct synthetic profiles. Results of the deconvolution technique from the left (red) and from the right (blue) are similar to the synthetic profiles (black). The profiles are centered between chord 11 and chord 12 to simulate a pinch not centered in the machine. Random noise has been added to the integrated, instrument broadened data at levels consistent with the experimental noise.

This technique is used to find the local plasma parameters of spectra measured with the ICCD spectrometer. Measuring the spectral intensities through twenty chords during one time interval ensures the impurity content and location of the plasma are the same for all twenty spectral intensities. This method only works when the edges of the Z-pinch are seen. An IDL code, *find_iccd_shell_param_SYIAH.pro* Sec. C.4.3, is used to find local plasma

parameters. A Gaussian fit is done to the line-integrated emissivity. If the entire FWHM of the Gaussian is within the chords, *find_center_SYIAH.pro* Sec. C.4.3 is used to find plasma parameters and the plasma location. The plasma parameters are then written to the tree.

Chapter 6

RESULTS FROM THE ZAP EXPERIMENT

Many experimental configurations have been used on the ZaP experiment. Changes have been made as the understanding of the Z-pinch has increased. The configuration of the electrodes has been modified to increase the lifetime of the Z-pinch. These changes involve a considerable amount of down time of the experiment, about a month. The capacitor bank has been modified to increase the half-cycle time of the current and increase the lifetime of the ignitrons. The magnitude of the current is controlled by varying the charge of the capacitor bank. The capacitor bank can be changed quickly, under a day, and the voltage can be varied from pulse to pulse. With each change, a study of reproducible discharges is conducted. New gas puff timings are found which generate reproducible Z-pinches, see Appendix B. Pulses with similar normalized m_1, Eq. 5.17, characteristics at $z = 0$ cm are then used to compare data taken at different times or locations. The results from four of the studies are shown in this Chapter.

The first study conducted on the experiment looked at the breakdown of the neutral gas. Asymmetries are seen in the initial breakdown of the gas. A uniform current sheet is then seen in the acceleration region. The rest of the studies, discussed later, show the velocity evolution of the Z-pinch. The next study, "the hollow inner electrode study," shows the results from the original experimental configuration. A stationary Z-pinch is seen during the quiescent period in the normalized mode data. The first measurements of a sheared flow Z-pinch were made during this study. Methane was added to the experiment to increase the brightness of higher ionization states of carbon. The results from this study, "the methane study," show that plasma is heating during the quiescent period. A differential velocity between the C ionization states is measured. The assumption that the impurities have the same parameters as the bulk plasma was verified shortly after this study by using He. The measured velocity profiles of He and C are similar. The last study, "the nose cone

study," uses lower energy pulses to monitor the time evolution of the velocity profile. The local velocity evolution of C III is measured throughout the quiescent period. The results presented in this Chapter will be discussed in the following Chapter. Each of these sections describes the experimental configuration, the bank configuration and charge, and initial the gas fill. The results from the acceleration region are then presented. Measurements of the Z-pinch characteristics are then shown. The velocity profiles are at the end of each section.

6.1 Breakdown of the Neutral Gas

On a typical plasma pulse the breakdown of the neutral gas cannot be studied. Instead data from the acceleration of the plasma are compared to breakdown studies initially performed on the ZaP experiment. The breakdown of the gas was studied by operating without the outer electrode endwall. Capacitor bank configuration 1 (see Table 4.1) was used to try to end the current before the plasma made it out of the outer electrode. The bank voltage was limited to 5 kV for this study. Only the inner gas puff valve was used for this study. The Imacon was used to record images of the optical emission as the gas broke down and the plasma as it formed. Shown in Fig. 6.1 is the most uniform breakdown of the neutral gas recorded. The photos are taken through the port on the endflange of the vacuum chamber. The dark circle in the center of the images is the inner electrode. Gas from four of the eight neutral gas injection broke down on this pulse. Typically, gas from only one or two of the injection ports initially breaks down. A uniform current sheet forms behind the filament and is accelerated out of the annulus, Fig. 6.2. The uniform ring inside the inner electrode is evidence of a uniform plasma sheet. The m_1 mode amplitude at the exit of the acceleration region agrees with the photos. The asymmetries in the breakdown process have not adversely affected the experiment since a uniform current sheet is established in a short time.

The impurity content of the plasma was studied with the CCD spectrometer. The metal content of the plasma is small even though the current is always connected to electrodes. The gas puff timing affects the impurity content, shown in Fig. 6.3. The metal content increases when the initial conditions starve the plasma of charge carriers. The uniformity

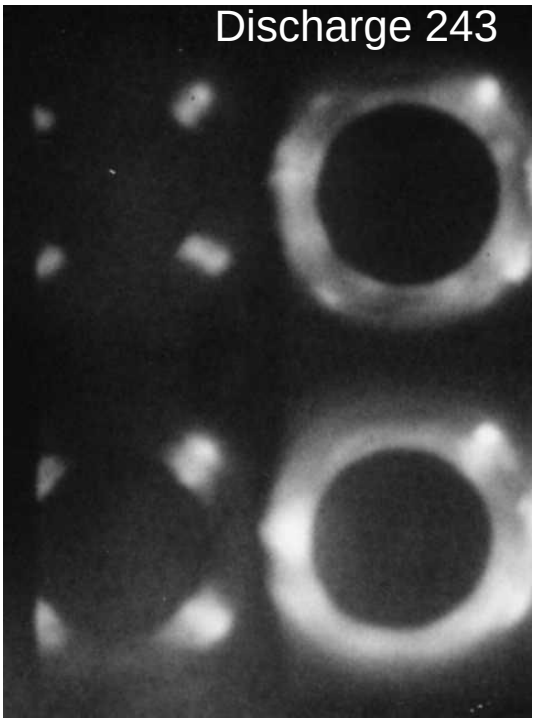

Figure 6.1: The Imacon records images of the plasma of the plasma as it breaks down. Pulse 243 is the most uniform breakdown measured on the ZaP experiment. Typically only one of the eight possible regions of high density breaks down.

of the breakdown also increases with earlier gas puff timings. Since one of the requirements in the ZaP experiment is to study fusion grade plasmas, it is important that appropriate measures are taken to keep the impurity content low.

6.2 Hollow Inner Electrode Results

After the breakdown study was completed, the outer electrode endwall was installed. The bank voltage was slowly increased. Sets of gas puff timings were found, which generated reproducible Z-pinches. The results shown from this experimental configuration are from January through October, 2000. The plasma was made with the hollow inner electrode, the original outer electrode and the solid end wall. Hydrogen was used as the gas for this configuration. The IDS data are taken in March, 2000. The inner gas puff valve was triggered

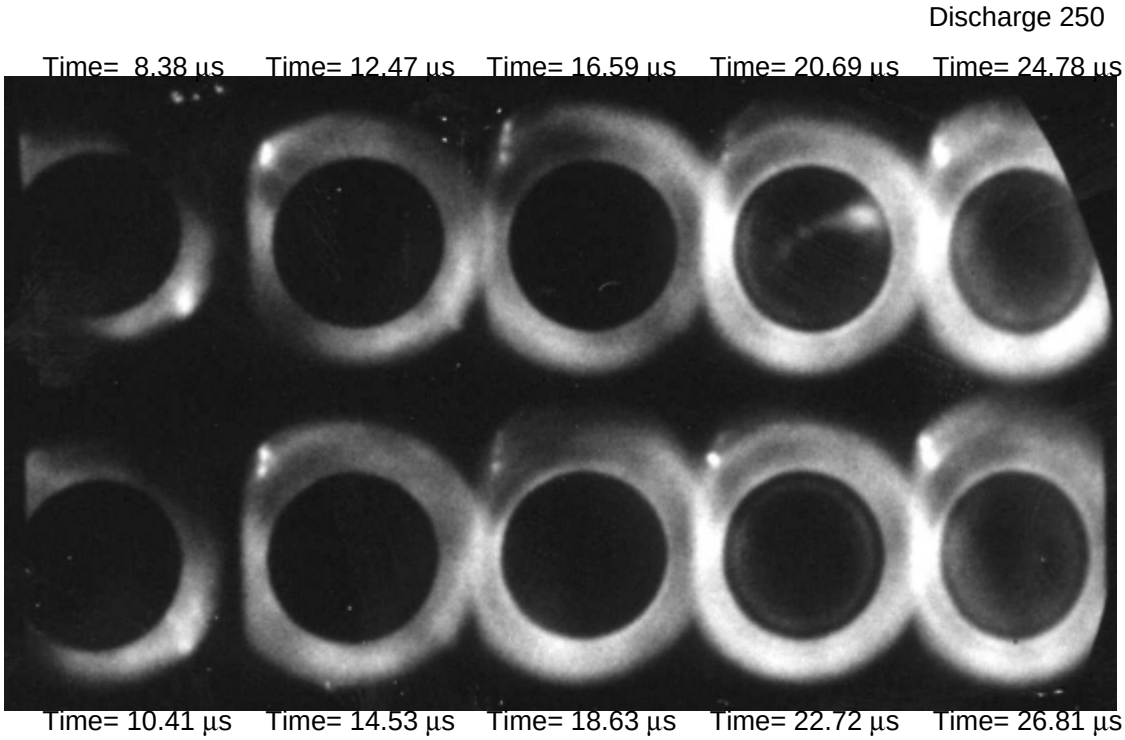

Figure 6.2: Asymmetric breakdowns are normally measured. Initially the breakdown is at opposite sides of the inner electrode. As the filaments travel down the acceleration region a uniform current sheet is formed. At 22.72 μs the halo inside the inner electrode shows a uniform plasma sheet attached inside the inner electrode.

at -1.75 ms. The two original outer gas puff valves were triggered at -2.25 ms. Capacitor bank 1 was charged to 9 kV for these pulses. The results from the ICCD spectrometer are from July, 2000. The gas puff timings were -2 ms for the inner gas puff valve and -2.25 ms for the outer gas puff valves. Capacitor bank 2 was charged to 8 kV for these pulses.

The current and voltage waveforms for the two banks are shown in Fig. 6.4. The initial currents and voltages are similar for the two banks until 30 μs. During this time the characteristics of both plasmas should be similar. The results in this section are from the first 40 μs, when the waveforms are similar. Data from the other diagnostics are used to show the similarities in the two configurations, IDS and ICCD spectrometer, during this time.

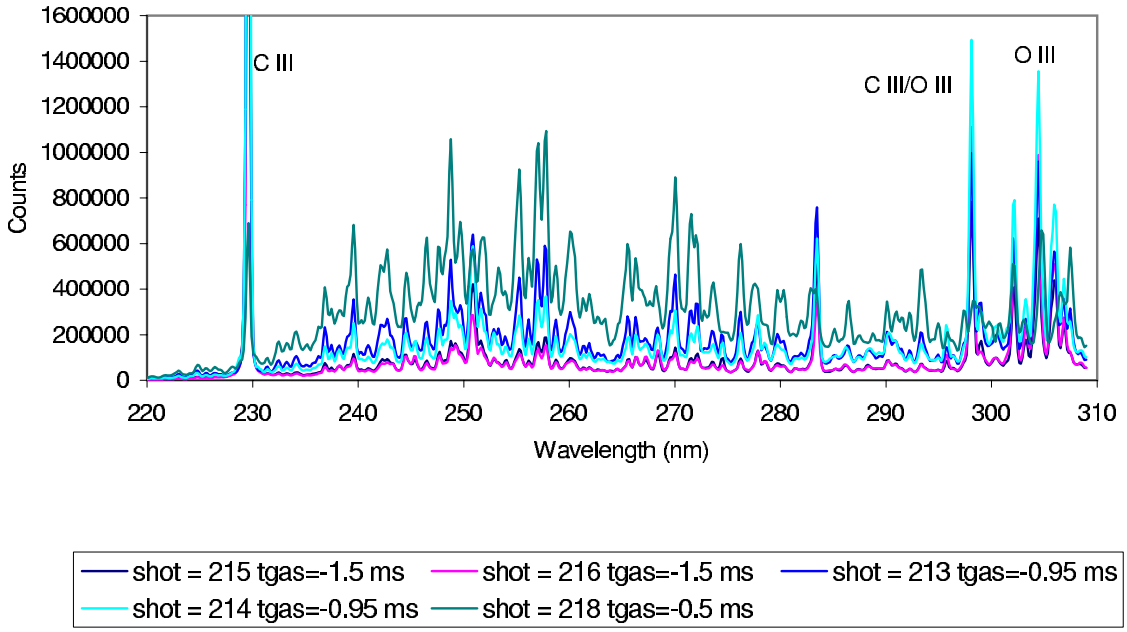

Figure 6.3: The spectra of the Z-pinch was taken with the CCD spectrometer taken during the breakdown studies. Most of the spectra from 230 nm to 290 nm is line radiation from tungsten. The spectra are from pulses with different gas puff timings. The tungsten content decreases as the gas puff timings are moved earlier in time.

6.2.1 Acceleration Region

The plasma is formed at the gas injection region of the inner electrode. After the current sheet forms, it is accelerated towards the assembly region by the $\mathbf{J} \times \mathbf{B}$ force. The current increases during the acceleration phase, shown in Fig. 6.4. The position, velocity, and acceleration of the current sheet at the outer electrode are measured by the axial magnetic probe array. Shown in Fig. 6.5 are magnetic fields at 2 μs intervals for the two configurations. The negative magnetic fields from 8 μs to 12 μs in the top plot are from filaments of current. The location of the leading edge of the current sheet at the outer electrode is where the magnetic field begins to increase from zero. The location where the field begins to remain constant is the trailing edge. The current sheet exits the acceleration region earlier for the first configuration's pulses. The gas is puffed later and the current rises faster than on the second configuration's pulses.

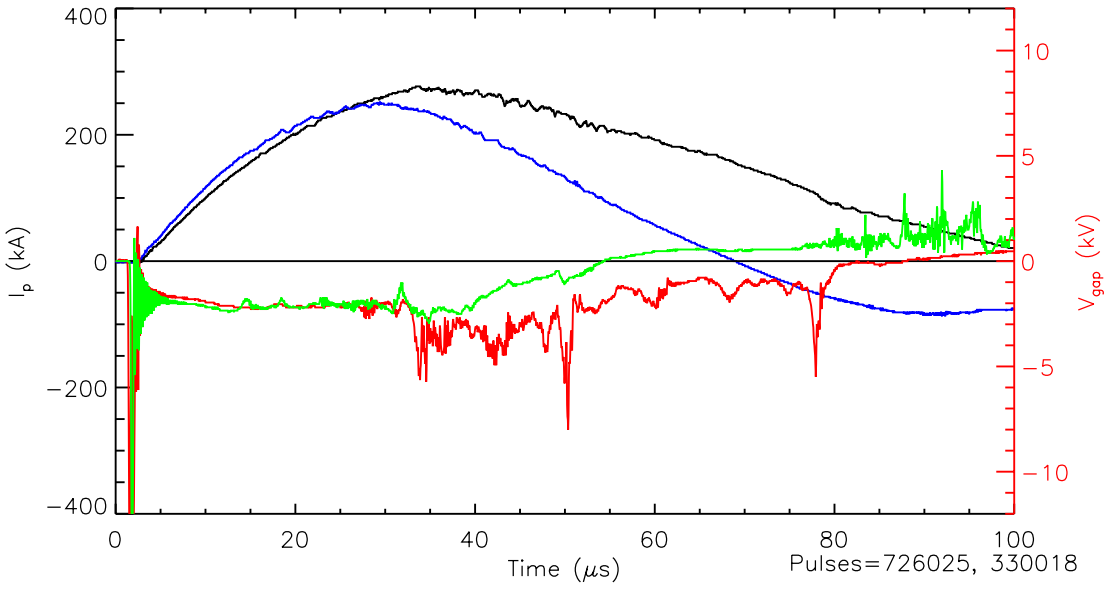

Figure 6.4: The initial plasma current (black) and electrode voltage (red) for pulse 726025 are similar to the plasma current (blue) and electrode voltage (green) for pulse 330018. The waveforms change at 30 μs due to the lower capacitance of the bank on pulse 330018. The shape of the waveforms show the experiment is not a simple RLC circuit.

The position and velocity of the leading edge on the current sheet along the outer electrode are found with the magnetic fields, as described in Sec. 5.5.5. The velocity of the current sheet for the two configurations are shown in Fig. 6.6. The velocity of the current sheet is increasing for both configurations. A larger current sheet velocity is seen in the first configuration. Less gas is in the acceleration region for this configuration.

Current filaments are measured with the azimuthal magnetic probe array at $z = -25$ cm. Large mode activity seen during this time is due to asymmetries in the current sheet, shown in Fig. 6.7. Filaments of current, which exit the accelerator ahead of the current sheet, will cause the m_1 and m_2 modes to be larger than the m_0 amplitude at the exit of the acceleration region. The magnetic field ahead of the filament is positive on one side of the electrode and negative on the other. The average magnetic field, m_0, must be zero from Ampere's law. After the current sheet passes the array at $z = -25$ cm, the low normalized

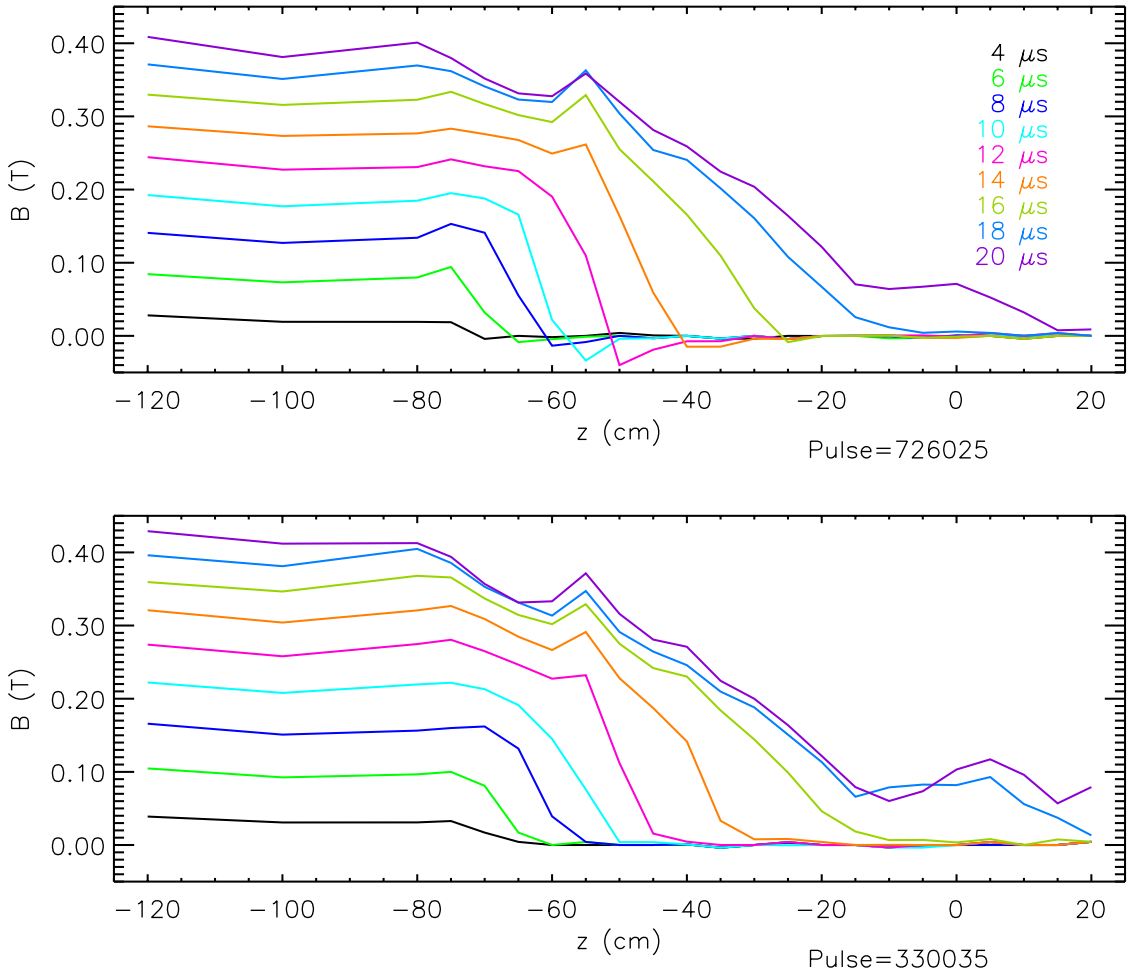

Figure 6.5: The magnetic fields measured by the axial array at 2 μs intervals show the acceleration of the plasma. Negative magnetic fields are measured ahead of filaments, when they are near the axial array.

mode amplitudes show the current is located on the z axis. The current then stays centered in the inner electrode, validating the assumptions used to calculate the current density in the accelerator.

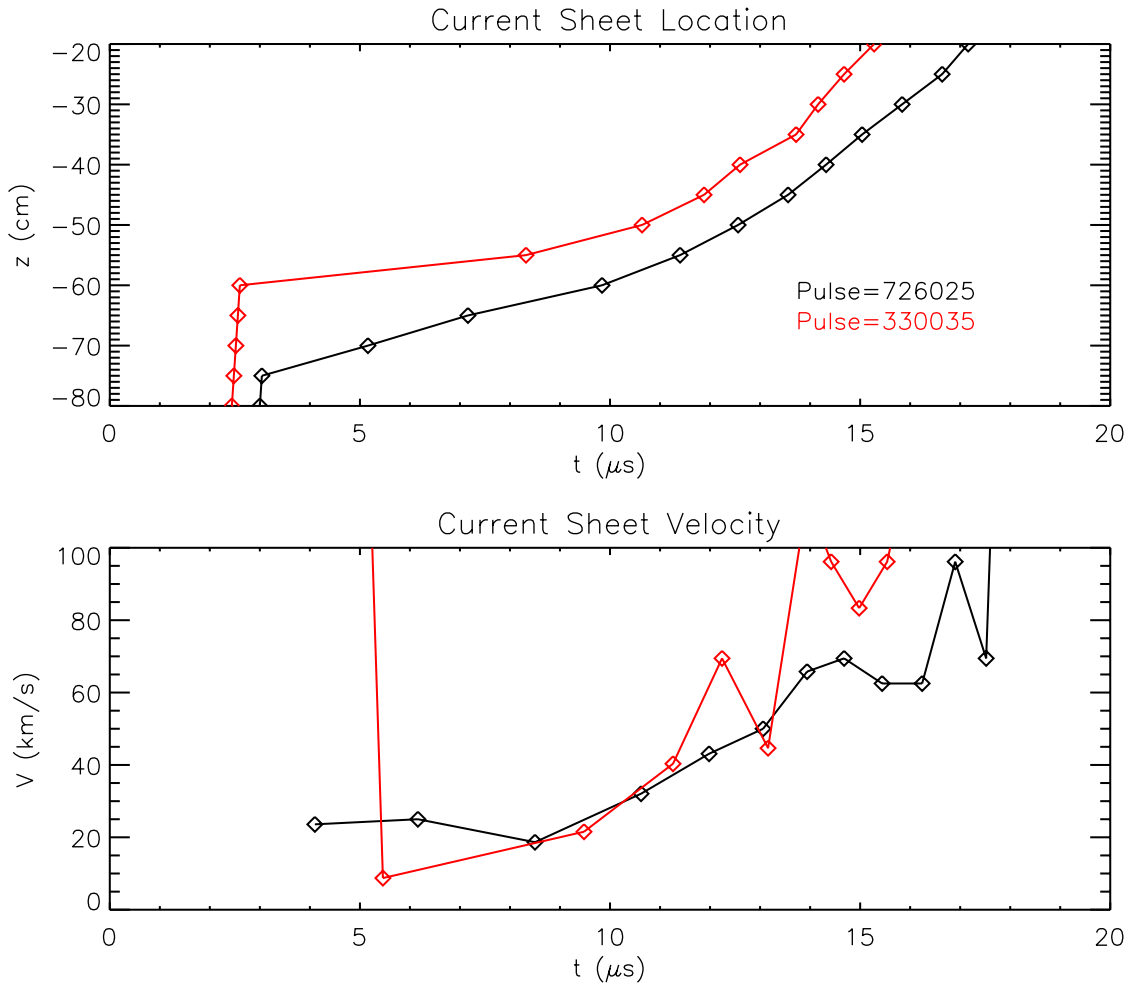

Figure 6.6: Shown are position (upper graph) and velocity (lower graph) of the leading edge of the current sheet along the outer electrode. The pulse (black) from the first configuration is accelerated more than from second (red).

6.2.2 Assembly Region

After the plasma exits the acceleration region, a Z-pinch is formed. The measurements of the plasma properties are made at the midplane of the Z-pinch, $z = 0$ cm. Axial variations in the Z-pinch should be small at this location. A quiescent period is seen in the normalized magnetic mode activity. Photos of emission from the plasma show a stationary plasma during this time. The plasma density is peaked on the axis of the machine. IDS measurements

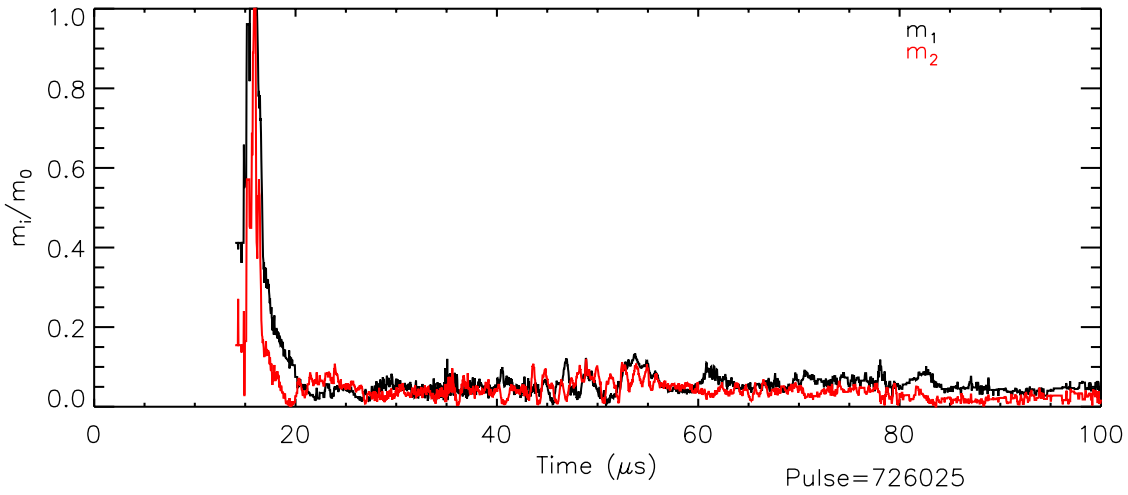

Figure 6.7: The normalized mode amplitudes at $z = -25$ cm are low for pulse 726025. The large mode amplitude from 15 to 18 μs is from current filaments which exit the accelerator ahead of the current sheet. From 20 μs on, the current is centered in the outer electrode at this position.

show the plasma is heating during the quiescent period. A sheared flow is measured with the ICCD spectrometer during the quiescent period.

Magnetic Field Measurements

The general character of the plasma pulse is described by the mode data. The average magnetic field and radial location of the Z-pinch are found with the azimuthal array of magnetic probes at $z = 0$ cm. The amplitudes of the Fourier components for both configurations are shown in Fig. 6.8. Initially, the average magnetic field, the m_0 component, is 0.1 T. It then rises to 0.25 T at 40 μs. m_0 is larger than the other components. The mode amplitudes have a similar character before 40 μs.

The radial location of the Z-pinch is found with the normalized mode data, Eq. 5.22. The normalized modes are shown in Fig. 6.9. Both configurations have similar normalized mode data for the first 40 μs. Large mode activity is measured early in time as the pinch is forming. Asymmetries in the current sheet cause this behavior. Shortly after formation, the normalized mode amplitudes decrease below an empirical value 0.2. The mode activity

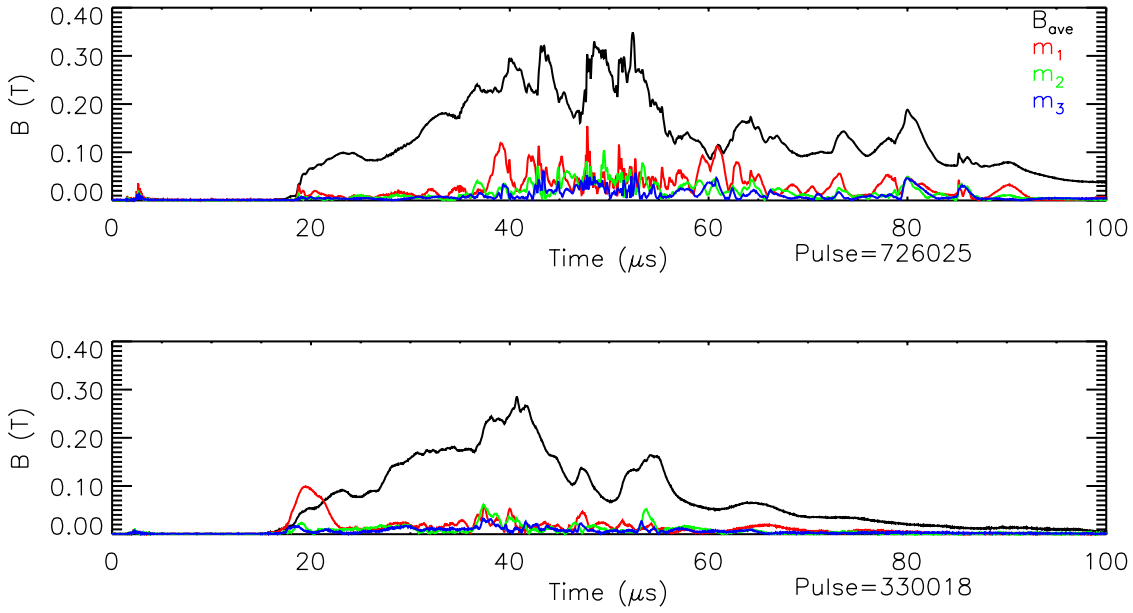

Figure 6.8: The magnetic mode amplitudes measured at the $z = 0$ array for the two configurations are similar during the first 40 μs. The average magnetic field at the wall varies from 0.1 T to 0.25 T during the initial stages of the Z-pinch. The magnitudes of the other modes are lower than the average magnetic field, m_0.

stays below this value for a period of time. The current in the Z-pinch is located within 1 cm of the axis of the machine during this time. The time when all the normalized modes are below 0.2 is empirically defined as the quiescent period. During the quiescent period, a Z-pinch is seen on the axis of machine. The amplitude and frequency of the normalized modes then increases. At the end of the quiescent period, instability structures are seen in the Z-pinch. Since the normalized mode data describes the behavior of the Z-pinch, it is included as the bottom plot on most figures.

Emission Measurements

The behavior of the normalized modes are consistent with the images of the emission from the Z-pinch. The size and shape of the Z-pinch is estimated with photos of the emission from the plasma. The Imacon fast framing camera is used to measure the emission from

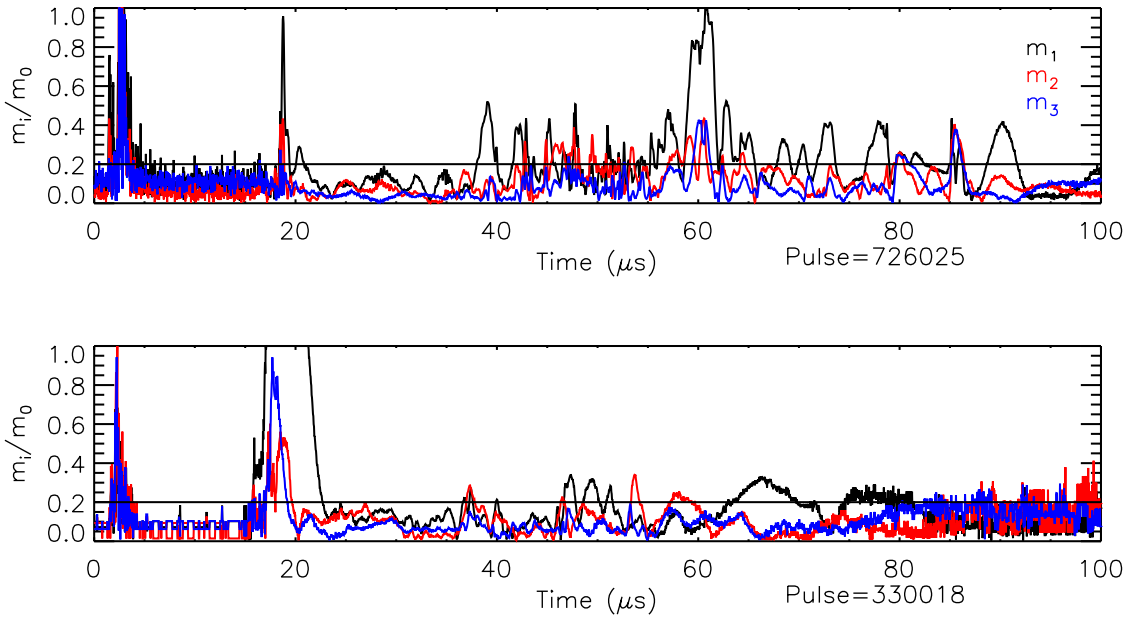

Figure 6.9: The normalized mode from a typical pulse has a quiescent period. The quiescent period is the time when all the normalized modes are below the horizontal line at 0.2. During this time a pinch is seen in the experiment. When the amplitude and character of the mode changes, instabilities are seen in the Z-pinch.

the plasma through one of the 2 inch holes in the outer electrode at $z = 0$ cm. Shown in Fig. 6.10 are four photos taken with different Imacon timings. The images on each photo are taken in 200 ns intervals. The upper left photo was taken during the formation of the Z-pinch. The upper right photo shows a stationary Z-pinch at $z = 0$ cm during the quiescent period. The size of the plasma seen in the photos are correlated with the density data. As the mode activity increases, perturbations are seen in the Z-pinch, lower two photos. The plasma can have an oscillation, lower left, or begin breaking up, lower right. The shape of the perturbations agrees with the normalized mode data.

Density Measurements

Density measurements of the Z-pinch are made with both chords of the interferometer. Both chords are located at $z = 0$ cm with different impact parameters, shown in Fig. 6.11. One

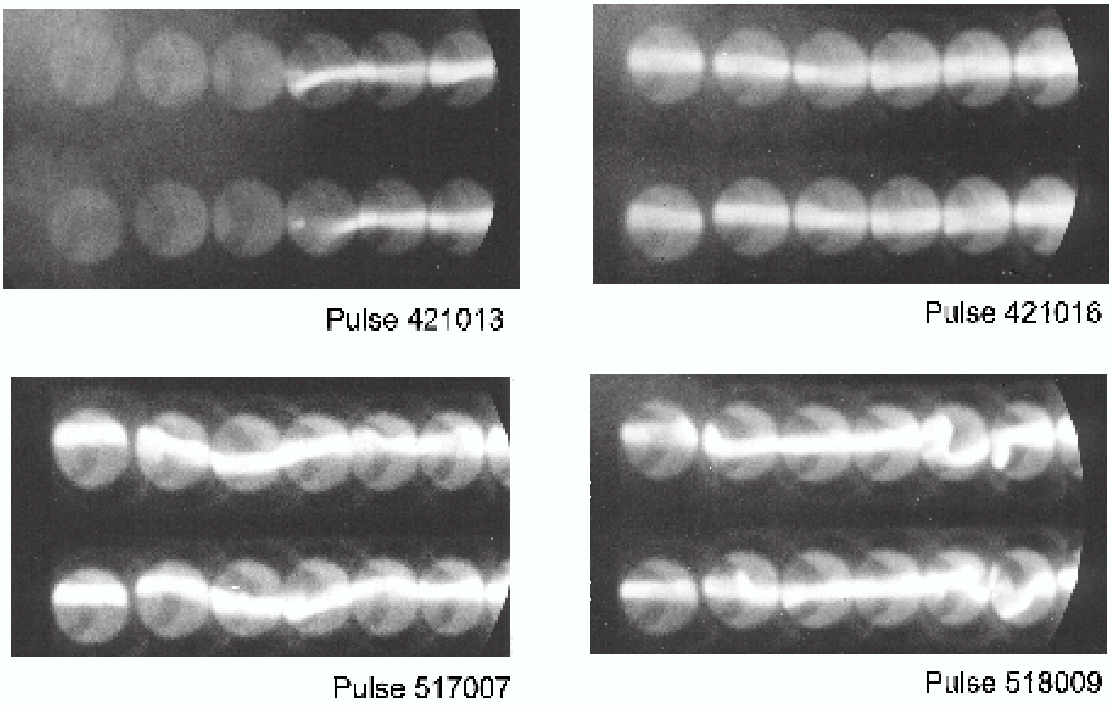

Figure 6.10: Photos of the plasma taken with the Imacon show a stationary Z-pinch in the machine. The images are taken through a 2 inch hole in the outer electrode. The order of the images is bottom to top, left to right. The time between successive images is 200 ns. The series of photos was taken during the IDS surveys. The upper left was taken shortly after formation. The upper right was taken of the quiescent period and the bottom two are taken late in the quiescent period.

chord is located through the center of the machine. The other chord is located 2 cm above the axis of the machine. The lengths of both chords through the outer electrode are the same. During the formation of the Z-pinch a uniform density is measured in the machine. The density is peaked during the quiescent period. The average density in the center of the machine is found with Eq. 5.39. The peaked density measurements are consistent with the photos, which show a Z-pinch in the axis of the machine during this time. During the increased mode activity, a uniform density is seen.

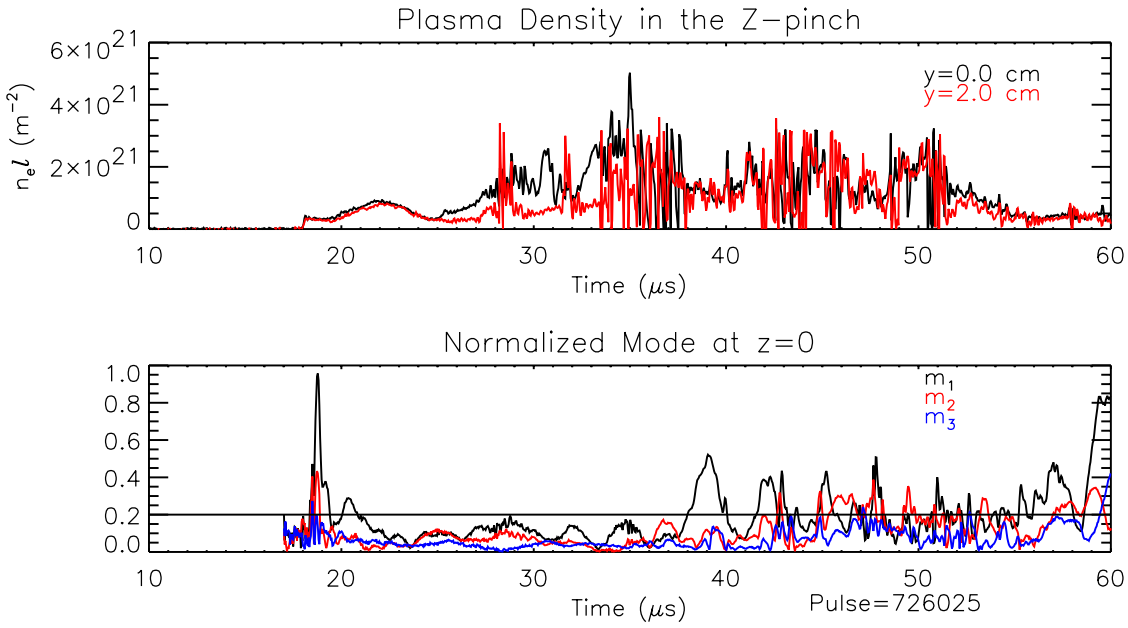

Figure 6.11: The chord integrated densities through the midplane show a Z-pinch. Both chords are located at $z = 0$ cm. Chord 1 is through the axis of the machine and chord 2 is located 2 cm above the axis of the machine. During the quiescent period, a peaked density is seen in the Z-pinch. The spikes in the data are from electrostatic noise on the signals.

IDS Measurements

The IDS system is used to measure the temporal evolution of the emission from different O states. The spatial structure of the Z-pinch is obtained by moving the fiber between similar pulses. The peak amplitude for different O states taken through various impact parameters is shown in Fig. 6.12. The intensity of the O IV line at 306.343 nm increases after the pinch forms and then decreases during the quiescent period. During the formation of the Z-pinch, the maximum intensity of the O IV is off the z axis. The peak intensity then moves to the center of the machine. As the intensity of the O IV is decaying, the intensity of the O V line at 278.1 nm increases on the axis. The O V intensity reaches a maximum, and then decays during which the O VI line at 381.135 nm increases in intensity. At this point the pinch becomes unstable and the amplitude of the O V and O IV lines increases. The evolution of the ionization states shows the plasma is being heated during the quiescent period.

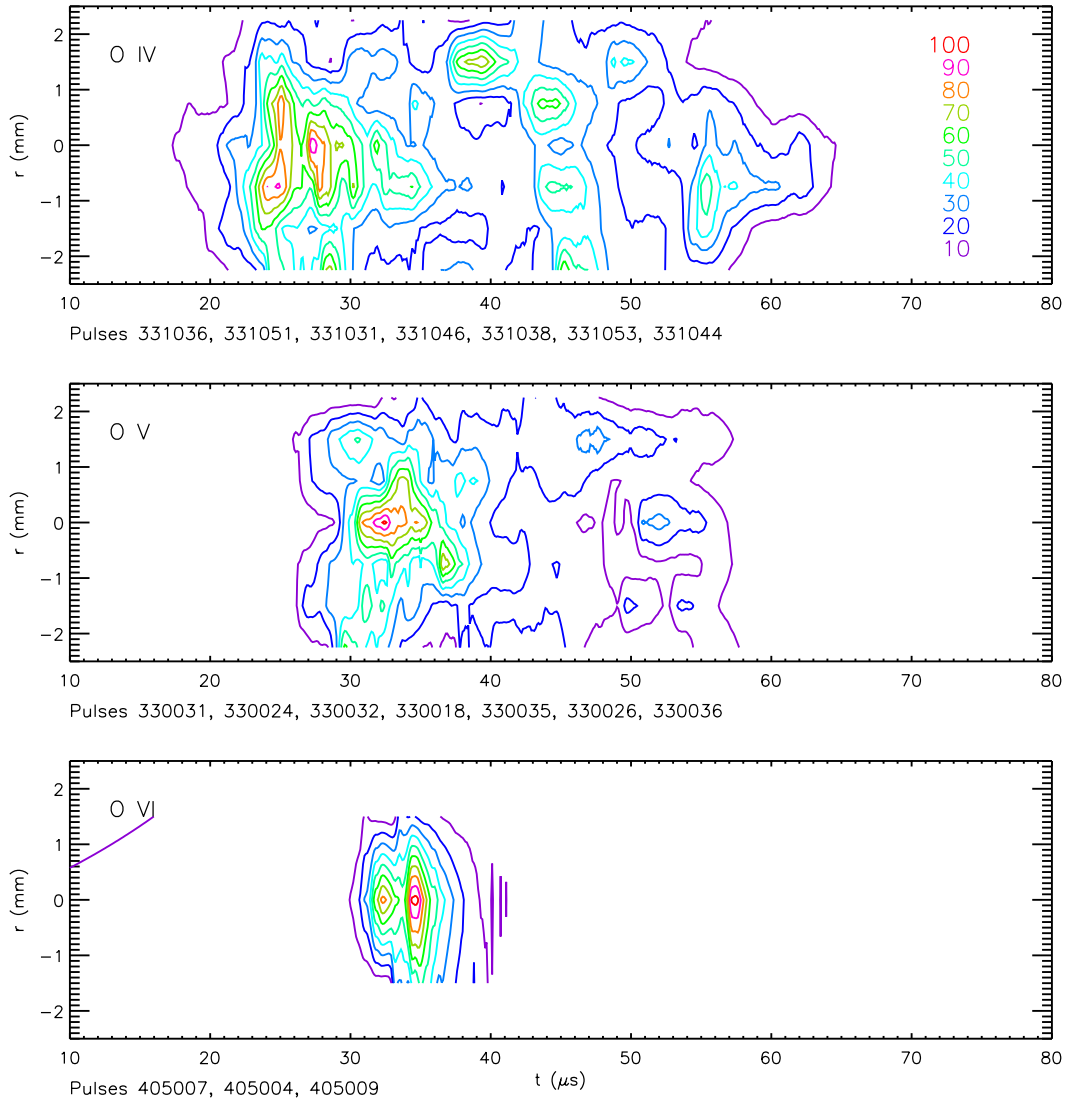

Figure 6.12: The spatial and temporal evolution of oxygen ionization states shows heating of the Z-pinch. The amplitude of the O IV emission is peaked off axis early in time. As the plasma heats, the amplitude of the O V increases on the axis of the experiment. Just before the mode activity increases, a maximum is seen in the O IV emission.

The temperatures measured with passive spectroscopy can be anomalously high. The evolution of the temperatures is shown in Fig. 6.13. Temperature measurements from IDS give an upper limit for T_i. Plasma dynamics in the Z-pinch can also broaden the line

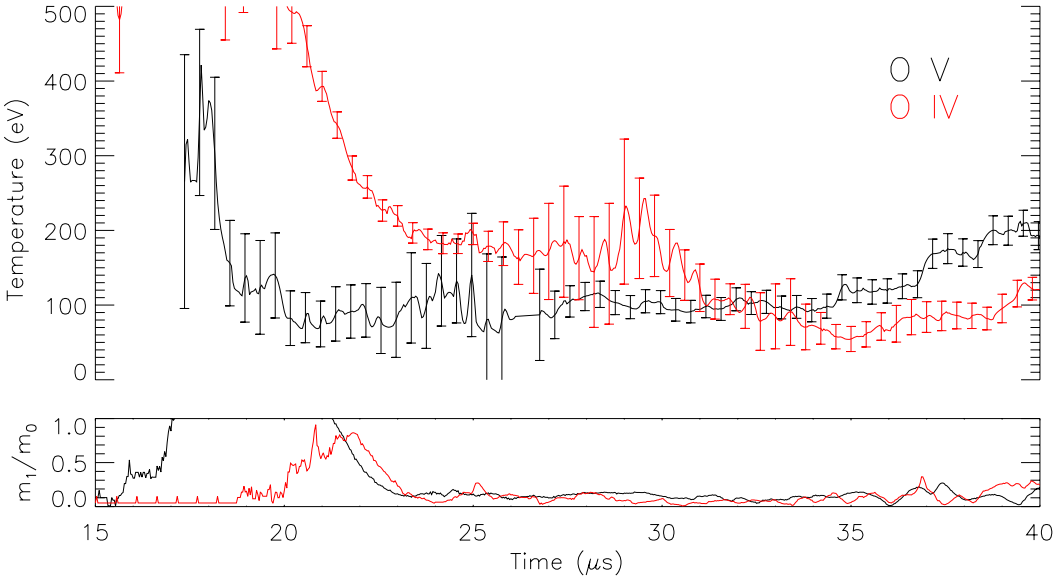

Figure 6.13: Ion temperature measured with the IDS instrument showing the ion temperature is 100 to 150 eV.

radiation. The actual width of the line is governed by

$$\triangle\lambda^2_{\text{obs}} = \triangle\lambda^2_{\text{thermal}} + \triangle\lambda^2_{\text{dynamics}} \tag{6.1}$$

where $\triangle\lambda_{\text{obs}}$ is the measured FWHM, $\triangle\lambda_{\text{thermal}}$ is the FWHM from Doppler broadening, and $\triangle\lambda_{\text{dynamics}}$ is the FWHM from plasma dynamics. During formation, the measured FWHM will be from more than doppler broadening. The current sheet is traveling towards and away from the telescope as the current sheet collapses. An axial velocity shear will widen the line when viewed from 35°. During the quiescent period, the temperatures of both ions are about 150 eV. The temperature from coronal equilibrium and pressure balance agree with this value.

The IDS spectrometer measures a high chord-integrated velocity during the quiescent period of the mode data, shown in Fig 6.14. As the velocity decreases below 100 km/s the mode activity increases. The O IV decreases faster than the O V. The higher temperature seen when the plasma is viewed at 35 degrees suggests a nonuniform velocity. The actual shear in the velocity cannot be measured with the IDS instrument. The variation in the

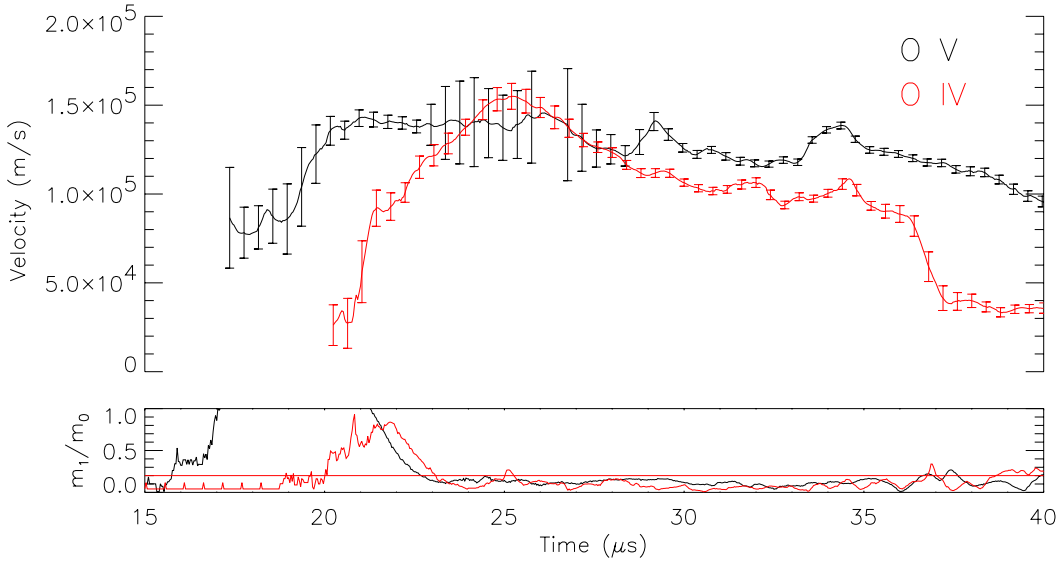

Figure 6.14: Ion velocities measured with the IDS spectrometer show the O V velocity is greater than the O IV velocity during the quiescent period. As the velocities decrease the mode activity increases.

plasma location from pulse to pulse does not allow for a quantitative measurement of the shear.

ICCD Spectrometer Measurements

The velocity shear is measured with the ICCD spectrometer. The evolution of the velocity profiles is found by taking repeatable plasma pulses and changing the ICCD spectrometer timing. The spectral intensities are measured over a 1 μs interval, shown as a blue trace in the normalized mode plot. When the emissivity is peaked, the local plasma parameters are found with the technique described in Sec. 5.7.5. The data were obtained one month and 10 days after the ICCD spectrometer was installed on the experiment.

The velocity is uniform during the formation of the Z-pinch. Shown in Fig. 6.15 is the C III velocity profiles measured during the formation of the Z-pinch. The local plasma parameters cannot be found since the chord integrated emissivity, shown in the top plot, is uniform. The velocity (middle plot) of the plasma is uniform during this time. The C III

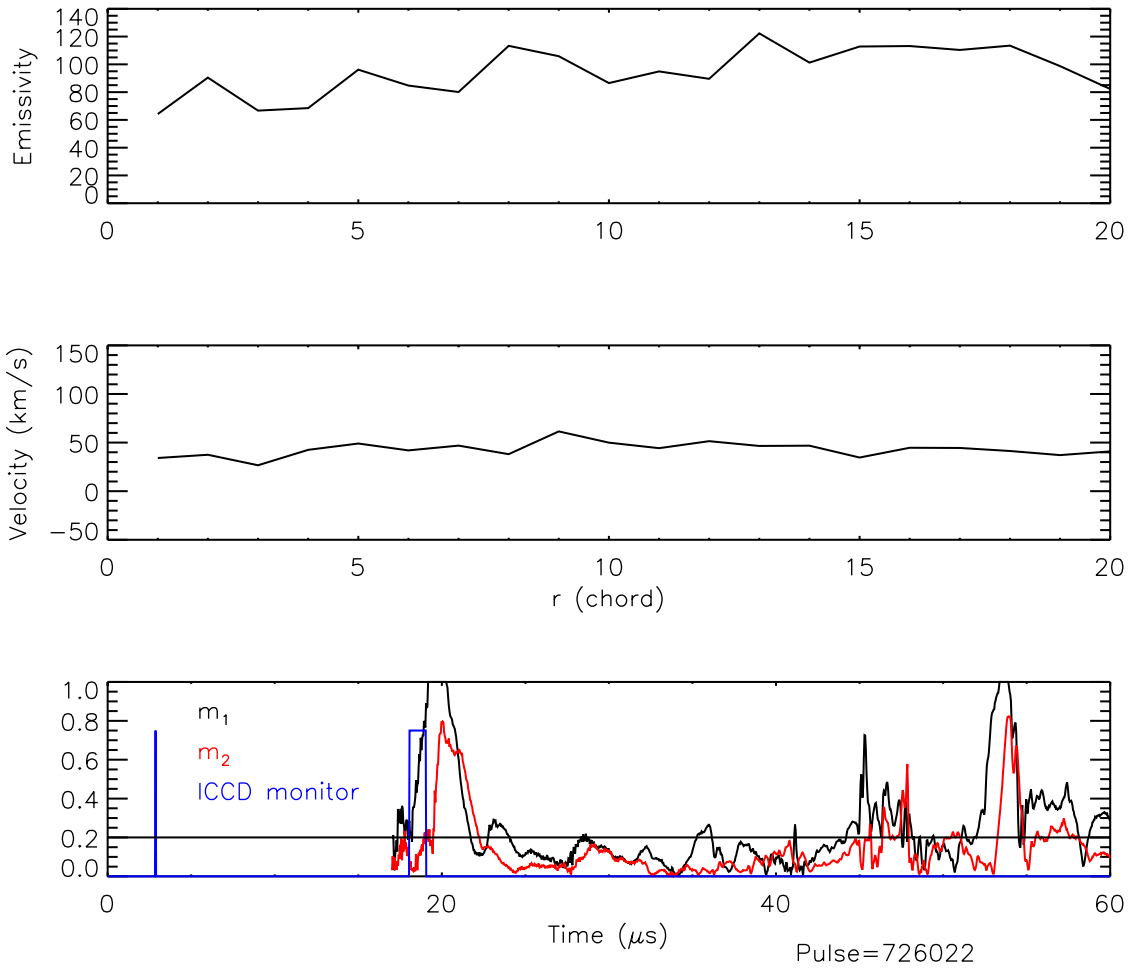

Figure 6.15: The chord integrated emissivity and velocity profiles are uniform during formation. The spatially broad emissivity profile precludes the use of the deconvolution technique. The plasma velocity during formation is uniform.

velocity is consistent with O IV and O V velocities during formation.

During the quiescent period, a velocity shear is measured in the Z-pinch. The velocity shear is shown in Fig. 6.16. The chord integrated profiles are shown in black. The red traces on the emissivity and velocity plots are calculated from the left side of the experiment. The parameters calculated from the right are shown in blue. This velocity measurement is one of the most publicized results of the ZaP experiment [57]. It shows a uniform velocity in the center of the Z-pinch. The velocity drops to zero in a thin layer on the edge of the plasma.

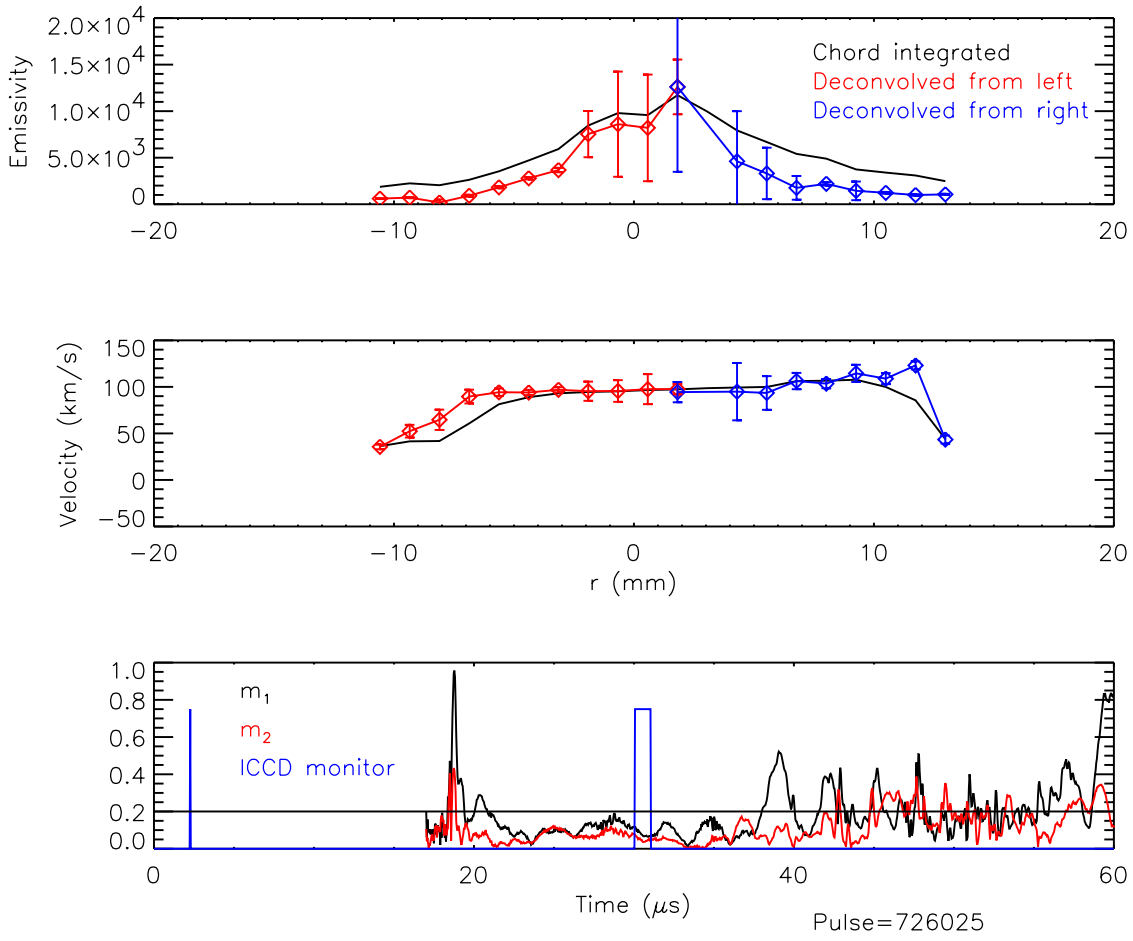

Figure 6.16: The plasma parameters measured with the ICCD spectrometer show a sheared flow Z-pinch. The emissivity profile is peaked on the axis of the machine during this time. The velocity shear is on the edge of the Z-pinch.

When the mode activity is high, the velocity and velocity shear are small, shown in Fig. 6.17. The local plasma parameters cannot be found since the emissivity profile is uniform. The velocity and the velocity shear are small during the increased mode activity.

6.3 Methane Results

The low impurity content in the plasma makes it difficult to measure the plasma parameters using higher ionization states. The intensity of the line can be increased by doping the

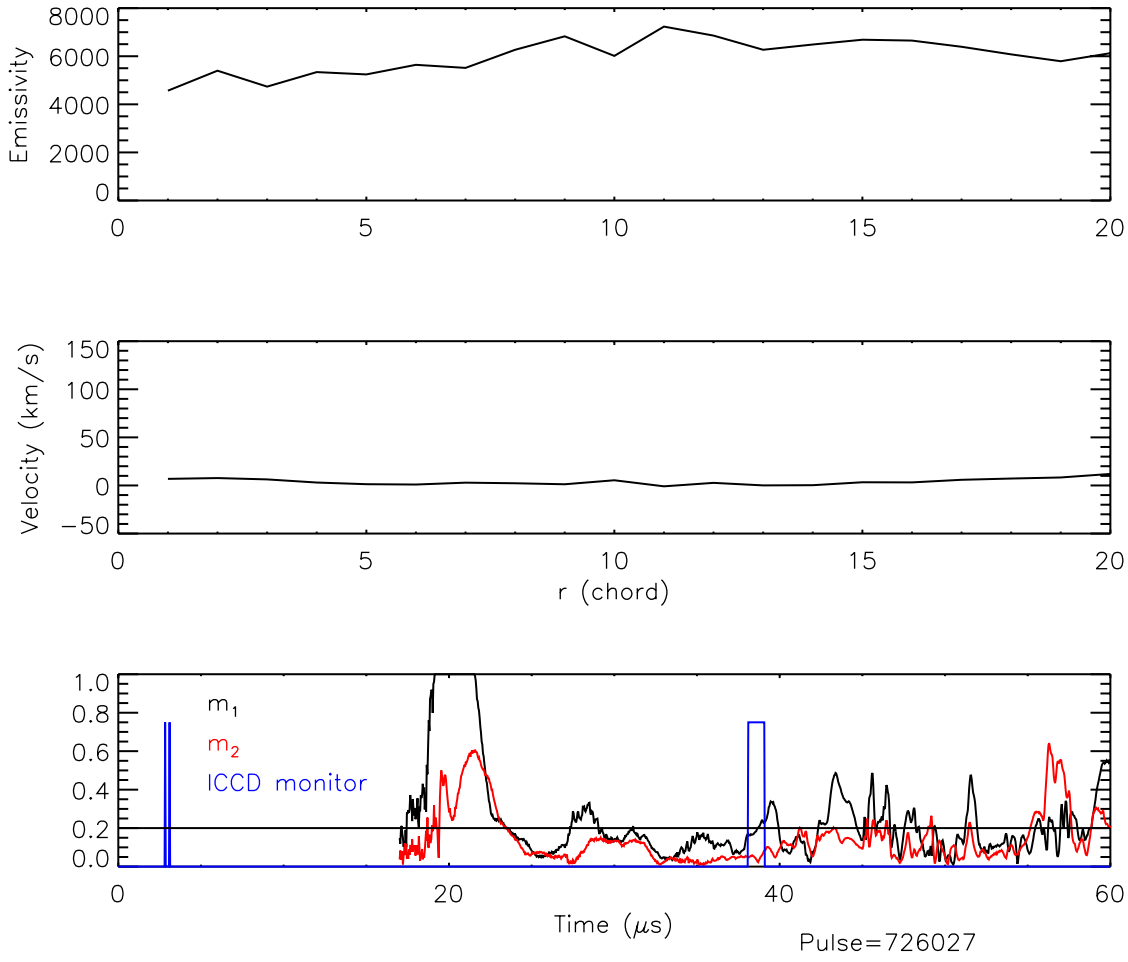

Figure 6.17: The velocity is low during increased mode activity. The local velocities could not be calculated since the emissivity profile is broad at this time.

plasma with an impurity. During the study in September and October, 2002, methane was added to the fill gas to increase the intensity of the C IV and C V lines. The plasma was made with the inner electrode nose cone attached, the original outer electrode and the end wall with the hole. A mixture of 50% hydrogen, 50% methane by pressure was used for this configuration to increase the brightness of the C IV and C V lines. The gas puff timings were -1.77 ms for the inner gas puff valve and -0.8 ms for all eight outer gas puff valves. Capacitor bank 1 was charged to 9 kV for these pulses.

The current and voltage waveforms for the methane study are shown in Fig. 6.18. The

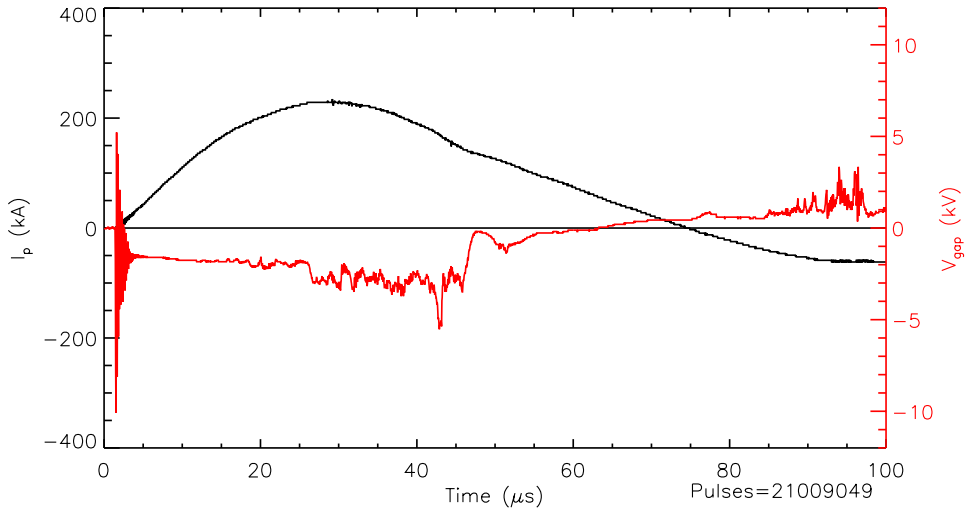

Figure 6.18: The plasma current (black) and electrode voltage (red) for pulse 21009049 are similar to the waveforms from the IDS study. The voltage waveform is decreases at 25–30 μs similar to the pulses from second configuration of the hollow inner electrode.

currents and voltages are similar to the IDS study. The oscillations in the waveforms are smaller than the pure H plasmas.

6.3.1 Acceleration Region

The behavior of the plasma is similar to the hollow inner electrode study. Two diagnostics are used to measure this process. The axial magnetic field array measures the location of the current sheet. The density of the current sheet is measured by the interferometer at two axial locations.

Magnetic Field Measurements

The plasma is formed at the midplane of the inner electrode. After the current sheet forms, it is accelerated towards the assembly region by the $\boldsymbol{J} \times \boldsymbol{B}$ force. The position, velocity, and acceleration of the current sheet at the outer electrode are measured by the axial magnetic probe array. Shown in Fig. 6.19 are magnetic fields at 2 μs intervals. The current sheet exits the acceleration region later than the hollow inner electrode studies. From the magnetic

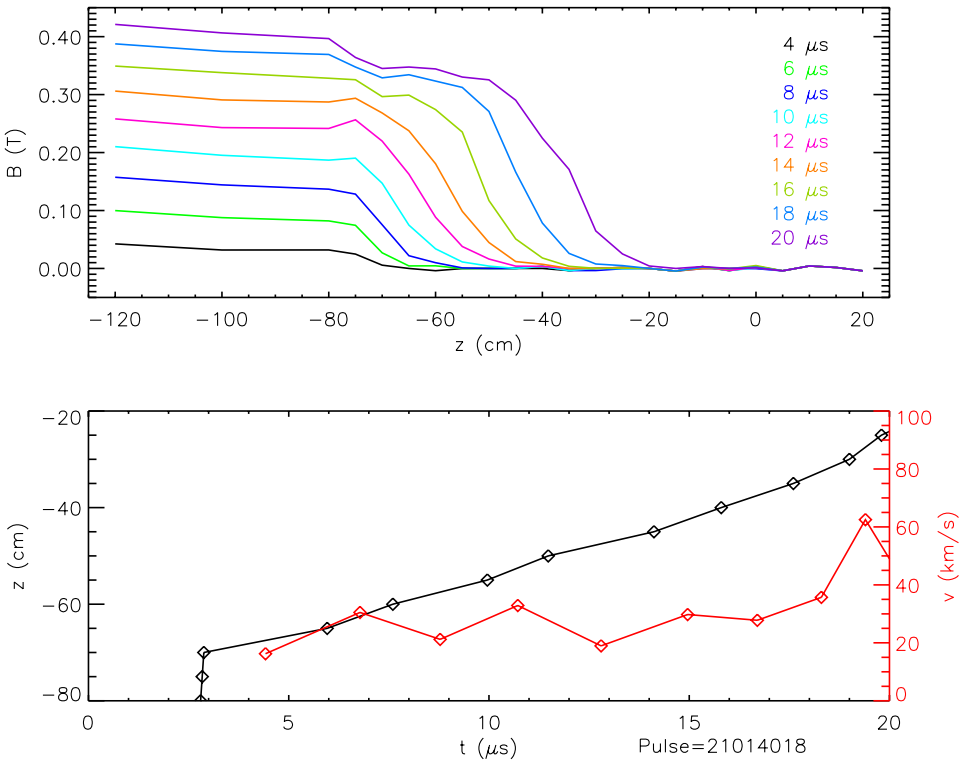

Figure 6.19: The position and velocity of the CH_4/H_2 current sheet are measured with the axial array. The magnetic fields in 2 μs intervals are shown in the top plot. The position (black) and velocity (red) of the leading edge of the current sheet along the outer electrode are shown in the lower plot. The velocity is lower than the H only current sheets.

fields the position and velocity of the leading edge of the current sheet along the outer electrode are found as described in Sec. 5.5.5. The acceleration of the current sheet is lower in the methane studies even though the magnitude of the magnetic field is the same as the hollow inner electrode studies. The density in the acceleration region is larger than the density for the hollow inner electrode studies.

Density Measurements

The density in the current sheet is measured through two axial locations with the interferometer. The line-integrated densities from the minimum impact parameters at two locations are shown in Fig. 6.20. The chord located at $z = -65$ cm (red) is located 10 cm downstream

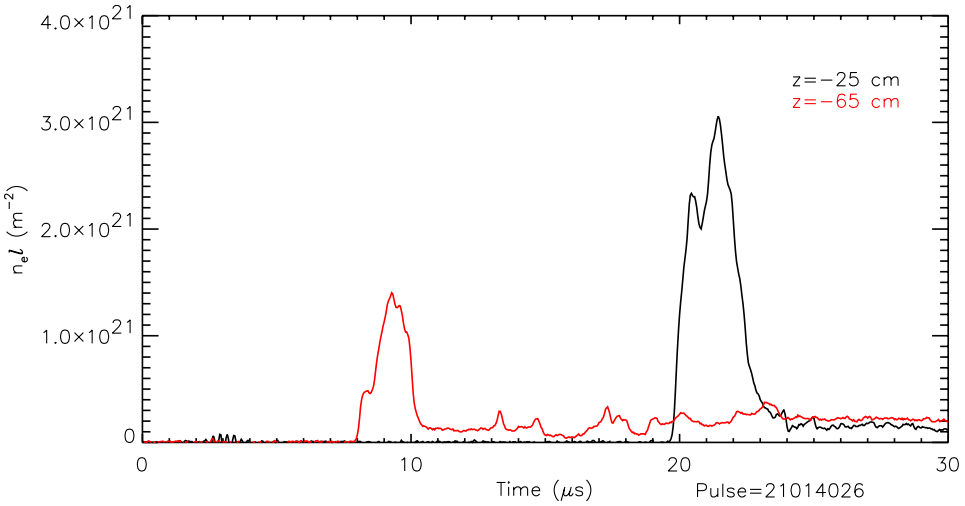

Figure 6.20: The density in the current sheet is measured with the interferometer. The density is increasing as the current sheet travels down the accelerator.

of the gas puff valves. The chord located at $z = -25$ cm (black) is located at the exit of the acceleration region. The spike in the density is the current sheet passing the axial location. The density does not return to zero after the current sheet passes the axial location. The line-integrated density is larger at the exit of the acceleration region. The neutral gas is ionized and entrained in the current sheet.

6.3.2 Assembly Region

After the plasma exits the acceleration region, a Z-pinch is formed. The measurements of the plasma properties are made at the midplane of the Z-pinch $z = 0$ cm. Axial variations in the Z-pinch should be small at this location. A quiescent period is seen in the normalized magnetic mode activity. The plasma density is peaked on the axis of the machine. The CCD spectrometer's measurements show the plasma is heating during the quiescent period. A sheared flow is measured with the ICCD spectrometer during the quiescent period.

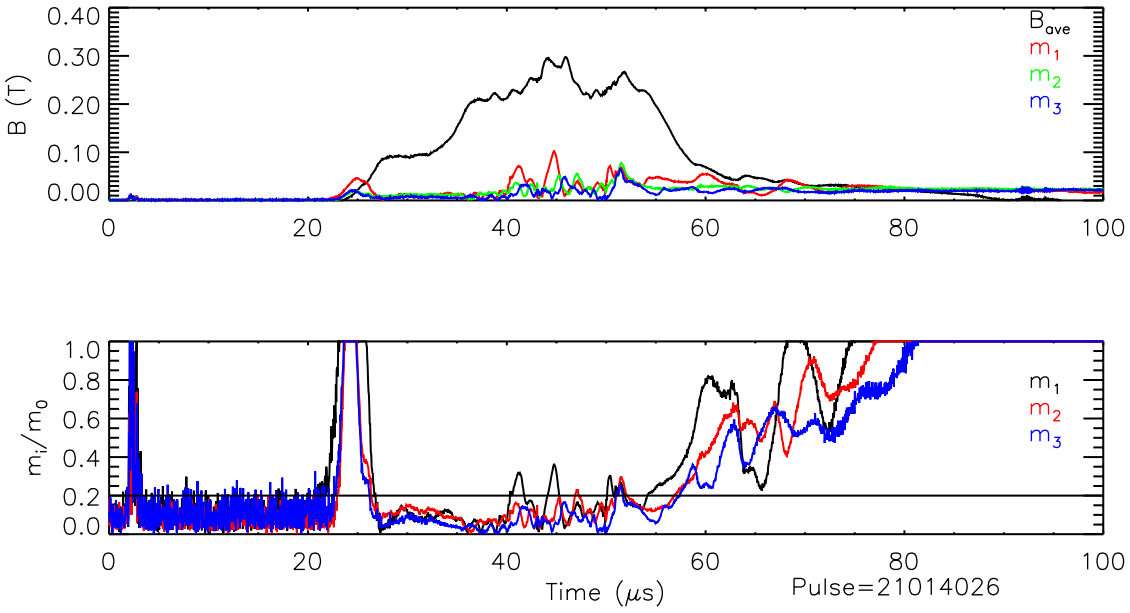

Figure 6.21: The magnetic mode amplitudes measured at the $z = 0$ cm array are shown in the top plot. The average magnetic field at the wall is 0.1 T as the Z-pinch forms. The magnitude of the magnetic field increases later in the quiescent period. The normalized mode is shown in the bottom plot. The quiescent period is similar to the hollow inner electrode results. The amplitudes do not return to zero late in time. Noise, during the increased mode activity, in the surface probe system causes the magnetic probe signals to return to a nonzero value.

Magnetic Field Measurements

The general character of the plasma pulse is described by the mode data. The average magnetic field and radial location of the Z-pinch is found with the azimuthal array of magnetic probes at $z = 0$ cm. The amplitudes of the Fourier components are shown in the top plot of Fig. 6.8. The average magnetic field, the m_0 component, is similar to the hollow inner electrode studies. The field is initially 0.1 T and then increases to 0.25 T. The normalized mode data are shown in the bottom plot. Both configurations have similar mode data for the first 40 μs. Large mode activity is measured early in time as the pinch is forming. Asymmetries in the current sheet cause this behavior. Shortly after formation, the normalized mode amplitudes decrease below 0.2. Few fluctuations of the normalized

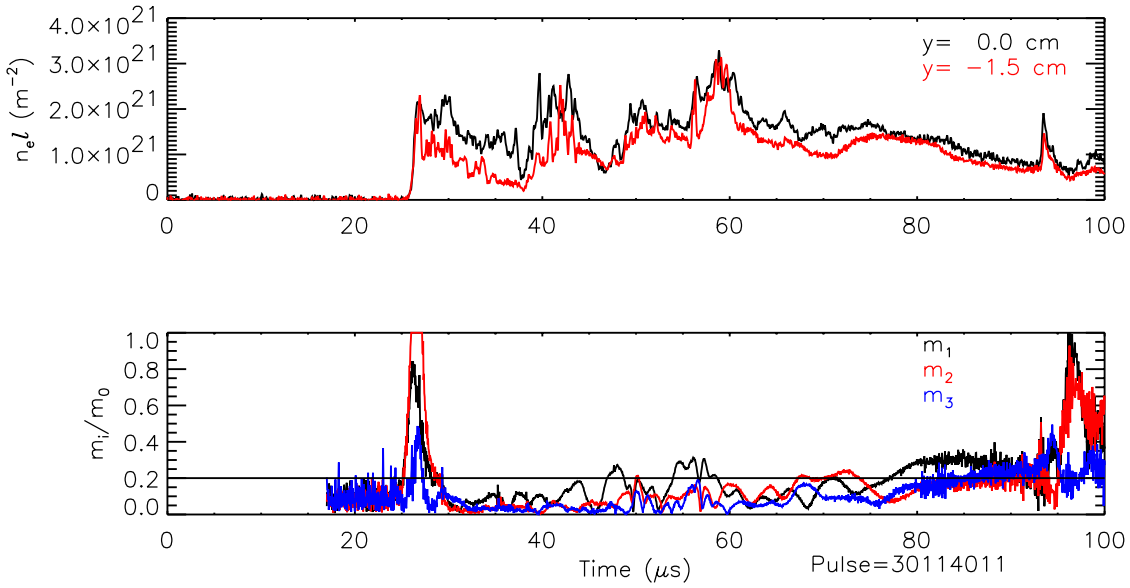

Figure 6.22: Shown are the chord integrated densities through $z = 0$ cm. Chord 1 is through the axis of the machine and chord 2 is located 1.5 cm below chord 1. During the quiescent period, a peaked density is seen in the Z-pinch.

mode amplitude are seen during this time. The plasma current is located within 1 cm of the axis of the machine. During the quiescent period, a Z-pinch is seen on the axis of the machine. The amplitude and frequency of the normalized mode amplitudes then increases. At the end of the quiescent period, the amplitude of the oscillations in the magnetic mode data is lower than the oscillations seen with the hollow inner electrode.

Density Measurements

Density measurements of the Z-pinch are made with both chords of the interferometer. The two chords are located at $z = 0$ cm. One chord is located at the center of the machine and the other chord is located 1.5 cm below the axis of the machine. The density measurements are shown in Fig. 6.22. During the formation of the Z-pinch, a uniform density is measured in the machine. The density is peaked during the quiescent period. The background density is higher on the methane study than on the hollow inner electrode study. During the increased mode activity, the difference between the two densities is lower.

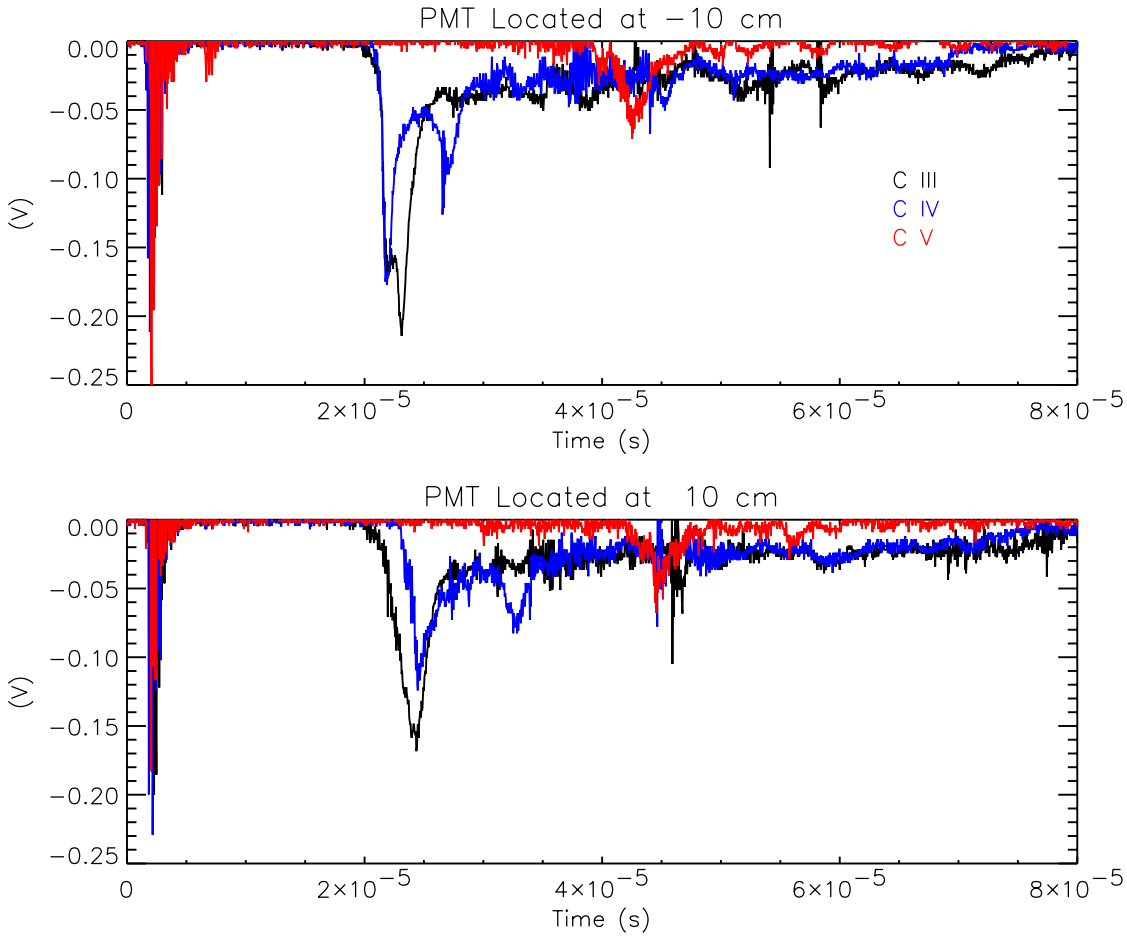

Figure 6.23: During the quiescent period the ionization state of C increases. The top plot is from the telescope located at $z = -10$ cm. The bottom is from the telescope located at $z = 10$ cm. The time of the peak emission of each ionization state increases with the ionization level.

Emission Measurements

Heating is seen in the Z-pinch during the quiescent period. The evolution of the ion species is evidence of plasma heating. The PMT on the CCD spectrometer is used to measure the evolution of different ionization states of C at two locations, shown in Fig. 6.23. The peak of the C III emission is during the formation of the Z-pinch. Later during the formation of the Z-Pinch, the C IV emission peaks. A second peak of the C IV is seen during the quiescent

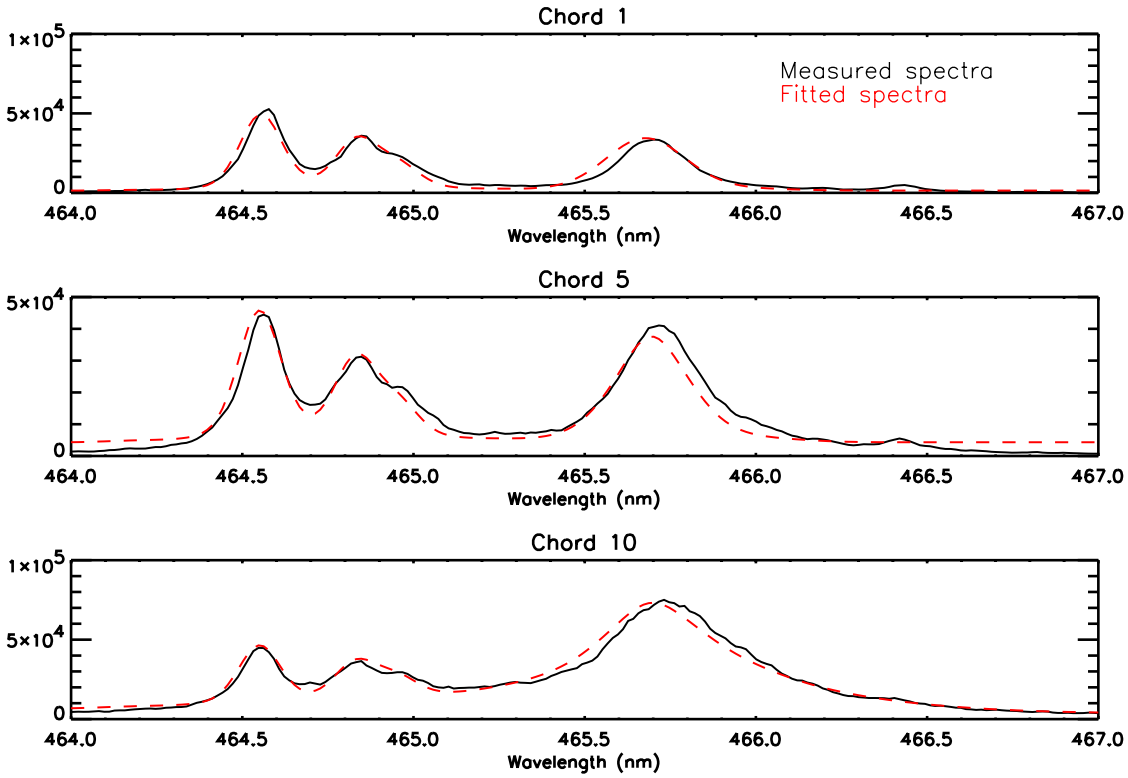

Figure 6.24: The spectra from C III and C IV around 465 nm is complicated. The lines from the two ionization states overlap each other in this region. The fitting routines are able to reconstruct the measured spectral intensities.

period. Before the end of the quiescent period, a peak is seen in the C V emission. The carbon burns through to a higher ionization state when the emission from higher ionization states increases in this manner. The increase of ionization states is consistent with coronal equilibrium.

ICCD Spectrometer Measurements

The velocity profiles of multiple ionization states of C are measured when methane is added to the experiment. The C III and C V lines are separated in the 229 nm region. The spectral intensities of the C III and C IV lines overlap in the 465 nm region, as shown in Fig. 6.24. Another Gaussian has been added to Eq. 5.73. The emissivity, velocity, and

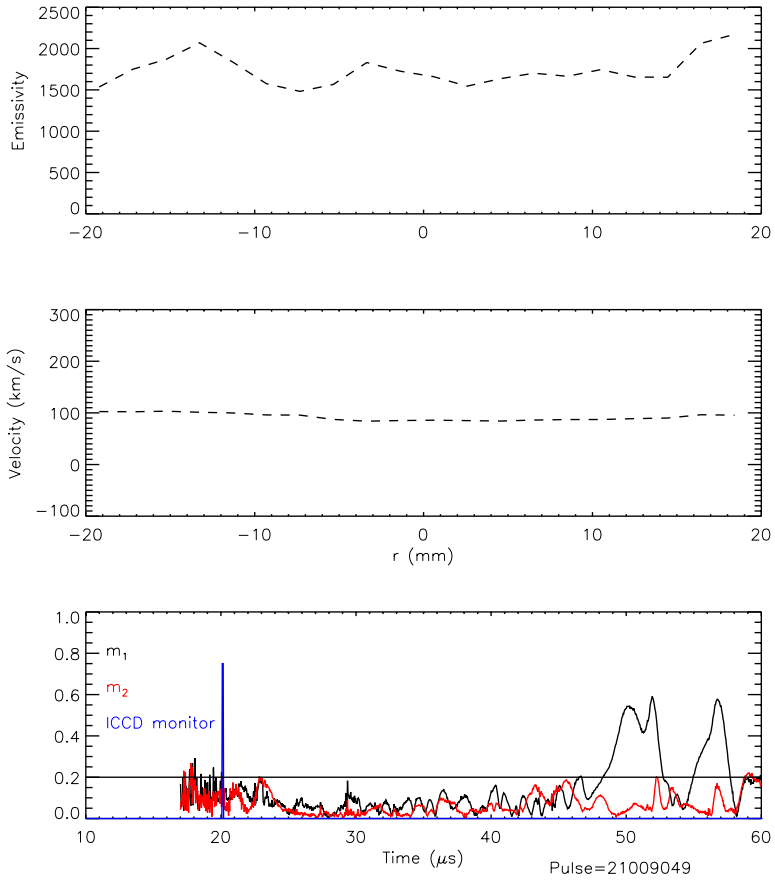

Figure 6.25: The chord integrated emissivity and velocity profiles are uniform during formation. The spatially broad emissivity profile precludes the use of the deconvolution technique. The plasma velocity during formation is uniform.

temperature of the C IV are independent fit parameters. The deconvolution technique is able to reconstruct the spectral intensities in 465 nm region.

The velocity is uniform during the formation of the Z-pinch. Shown in Fig. 6.25 is the C III velocity profile, measured during the formation. The local plasma parameters cannot be found since the chord integrated emissivity is uniform. The C III velocity profile is consistent hollow inner electrode velocities during formation.

During the quiescent period, a velocity shear is measured in the Z-pinch. Multiple ionization states are used to measure the velocity shear during the quiescent period The

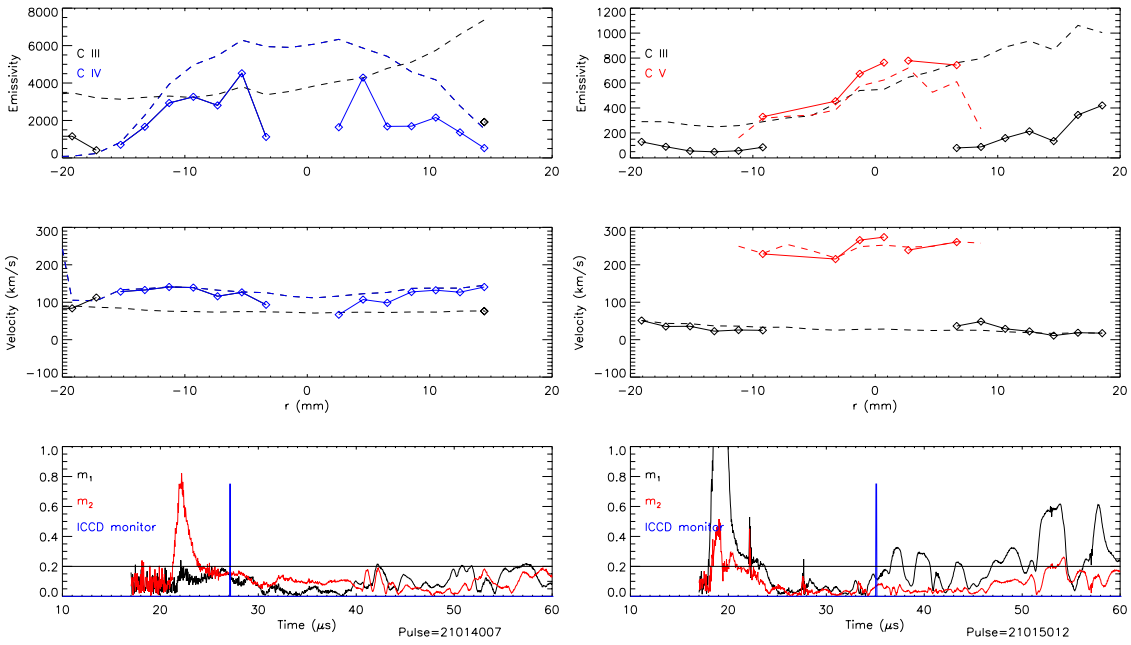

Figure 6.26: The plasma parameters measured with the ICCD spectrometer show a sheared flow Z-pinch. The chord-integrated profiles are shown as dashed lines. The local C III parameters are found using the geometric factors from the C IV or C V deconvolution. The velocity shear is on the edge of the Z-pinch.

velocity profiles of C III, C IV, and C V are shown in Fig. 6.26. The chord-integrated profiles are shown as dashed lines. The C III emission is from the edge of the Z-pinch. A hollow emission profile is seen in the C IV emission. The C V emission is from the center of the Z-pinch. The velocity shear is the difference of the velocities of the ionization species. The edge velocity, measured with C III is small. The higher ionization states show the velocity is greater in the center of the Z-pinch. The exact magnitude of the velocity shear is difficult to measure when multiple ionization species are used and the radial location of the C III is not well defined.

When the mode activity is high, the velocity and velocity shear are small, shown in Fig. 6.27. The velocity measure with C IV has decreased as the mode activity increases. A smaller velocity shear is present at this time. After the mode activity has increased, the velocity is not sheared. The emission profiles of C III and C V are similar during this time

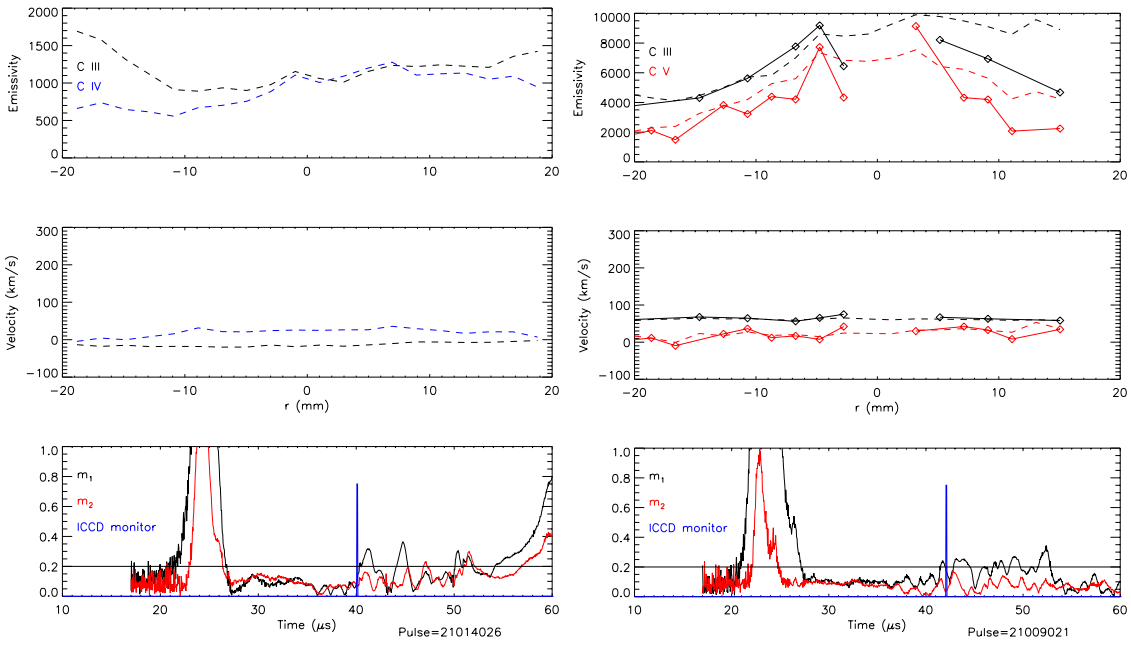

Figure 6.27: The velocity is low during increased mode activity. The local velocities for the C IV could not be calculated since the emissivity profile is broad at this time. The velocity of C V is less than the C III velocity.

and both velocities profiles are uniform. The C V velocity is smaller than the C III velocity. The velocity profile may have been uniform before this time.

6.4 Direct Plasma Velocity Measurements

The assumption that the impurities and bulk plasma have the same properties was verified by forming helium plasmas. The velocities of the bulk plasma and the impurities can be measured by viewing the spectral region around wavelengths of 468 nm. Plasmas were formed with helium in January and February, 2003. The gas puff timings were -2.4 ms for the inner gas puff valve and -1.0 ms for all eight outer gas puff valves. The rest of the configuration was the same as the methane study. The velocity of the bulk plasma, partially ionized He, could then be measured.

The spectral intensities of the C III at 464.7418 nm, C IV at 46.83 nm, and He II at

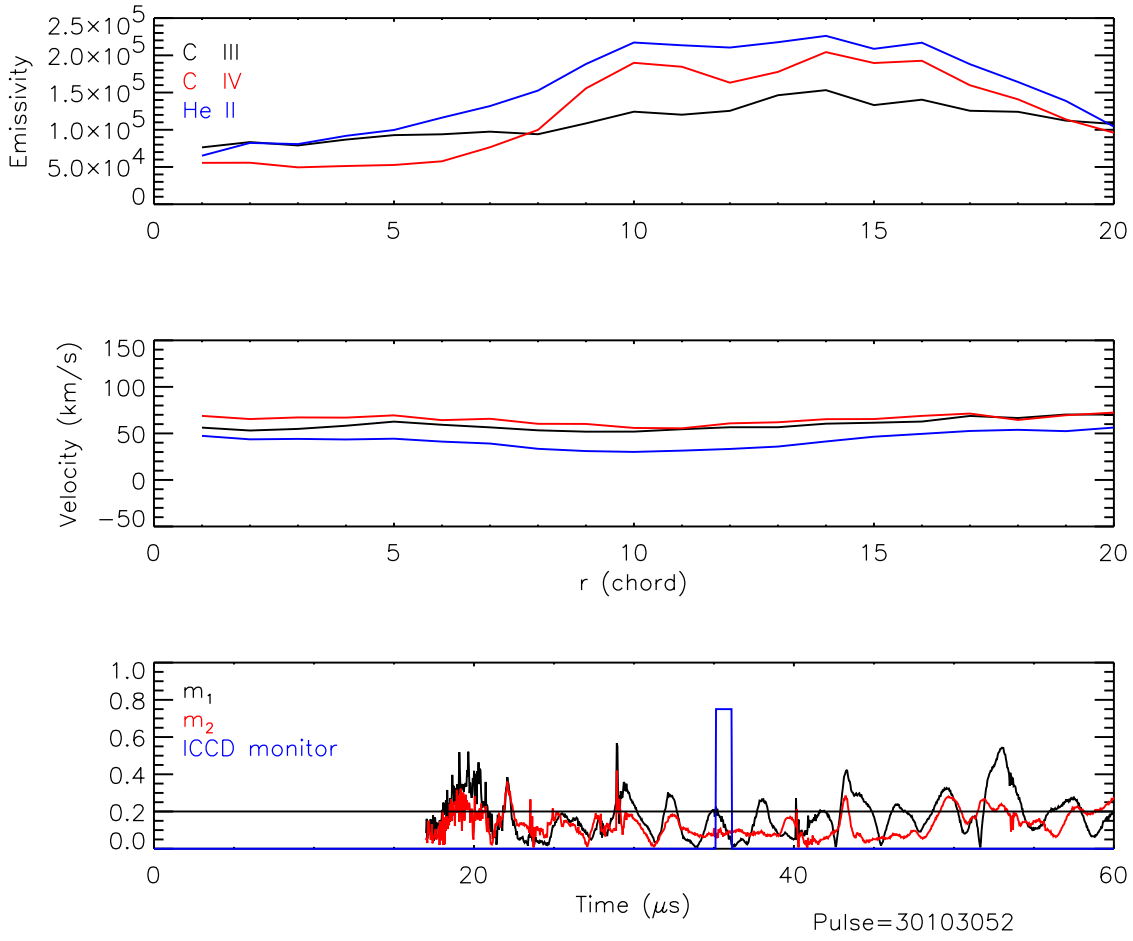

Figure 6.28: The velocity of the impurities are similar to the plasma velocity. The velocities measured with C III (black) and C IV (red) are similar in magnitude and shape to the velocity measured with He II (blue).

468.57 nm are measured on a single pulse with the ICCD spectrometer. The velocity of the bulk plasma, He II, is similar to the impurities, C III and C IV, shown in Fig. 6.28. The emissivity profiles are similar for all of the ions, top plot, and the velocity profiles, middle plot, are similar. The impurities and the bulk plasma have the same parameters.

The absolute velocity is measured with the "Y" fiber. The velocities measured at two angles to the z axis are shown in Fig. 6.29. The even chords, solid lines, view the plasma at 35° to the z axis. A velocity of 50 km/s is measured through these chords. The odd

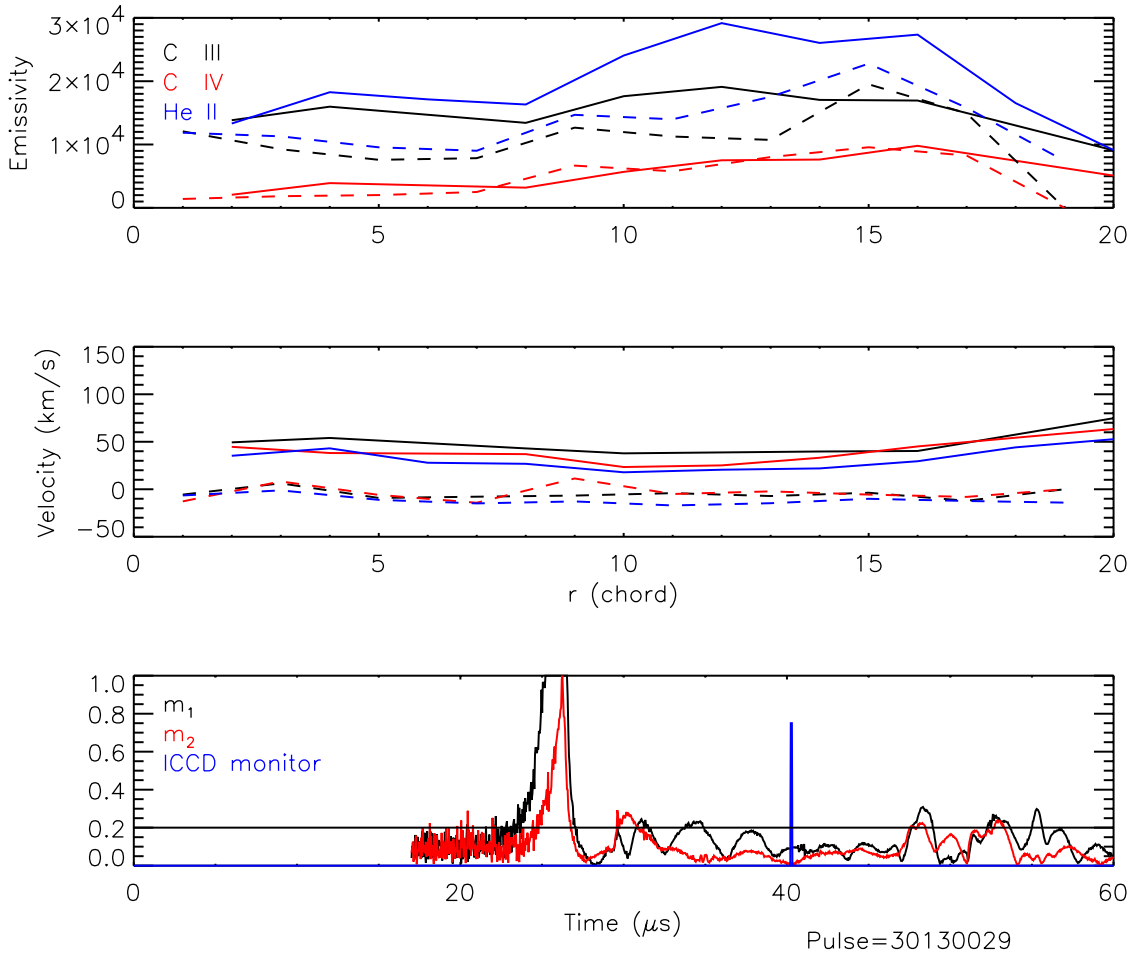

Figure 6.29: The "Y" measures the absolute velocity of the impurities and the bulk plasma. The fibers which view the plasma at 90° measure a low velocity (dashed line). The fibers which view the plasma at 35° measure a velocity of 50 km/s (solid line). The velocities measured with C III (black) and C IV (red) are similar in magnitude and shape to the velocity measured with He II (blue). The absolute velocity is the difference between two adjacent chords.

chords, dashed lines, view the plasma at 90° to the z axis. The measured velocity from these chords is about zero, as expected. Two of the assumptions used in the velocity calculations have been verified with the He plasmas. The impurities and the bulk plasma have the same parameters.

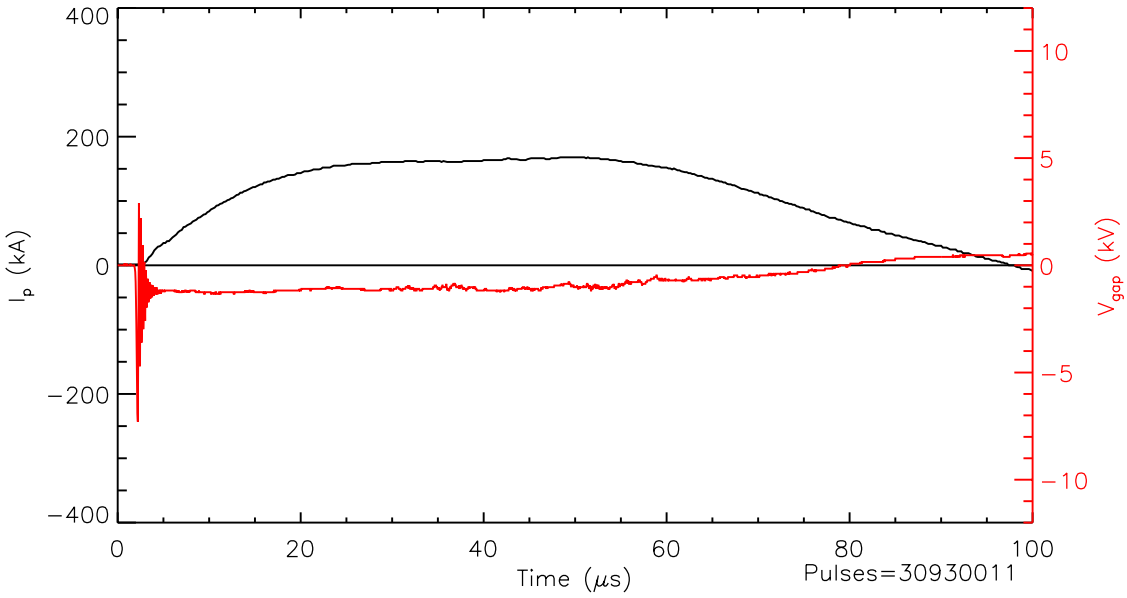

Figure 6.30: Shown are the plasma current (black) and gun voltage (red) for bank configuration 6 with a 5 kV charge. The driving wave forms are significantly different than the other configurations.

6.5 Nose Cone Results

When multiple ion species are used to find the velocity profiles, large uncertainties in the velocity shear arise. Decreasing the bank energy lessens the heating in the plasma. The temporal evolution of the velocity profile can be measured when one ion species is used. This experimental configuration was studied in September and October, 2003. The plasma was made with the inner electrode nose cone, the outer electrode with the extension and the end wall with the hole. Hydrogen was used as the gas for this configuration. The inner gas puff valve was triggered at -1.77 ms. Eight outer gas puff valves were used; four triggered at -0.8 ms and four triggered at -1.5 ms. Capacitor bank 6 was charged to 5 kV for these pulses.

The current and voltage waveforms are shown in Fig. 6.30. The current and voltage are constant for 30 μs with this configuration. The amplitude of the voltage and current are lower than other configurations. The voltage does not have an increase in amplitude as

seen in the other configurations. There are few spikes in the voltage with this experimental configuration.

6.5.1 Acceleration Region

The plasma is formed at the midplane of the inner electrode. The $\mathbf{J} \times \mathbf{B}$ then accelerates the plasma towards the assembly region. Two diagnostics are used to measure this process. The axial magnetic field array measures the location of the current sheet. The density of the current is measured by the interferometer at two axial locations.

Magnetic Field Measurements

After the current sheet forms it is accelerated towards the assembly region by the $\boldsymbol{J} \times \boldsymbol{B}$ force. The current is increasing during the acceleration phase as shown in Fig. 6.30. The position, velocity, and acceleration of the current sheet at the outer electrode are measured by the axial magnetic probe array. Shown in Fig. 6.31 are magnetic fields at 2 μs intervals. The current sheet exits the acceleration region later than the hollow inner electrode studies. From the magnetic fields the position and velocity of the leading edge on the current sheet along the outer electrode are found as described in Sec. 5.5.5. The velocity of the current sheet is similar to the methane studies, though the magnetic fields, are lower than the other studies.

Density Measurements

The density in the current sheet is measured through two axial locations with the interferometer. The line-integrated densities from the minimum impact parameters at two locations are shown in Fig. 6.32. The chord located at $z = -65$ cm (red) is located 10 cm downstream of the gas puff valves. The chord located at $z = -25$ cm (black) is located at the exit of the acceleration region. The spike in the density is the current sheet passing the axial location. The density does not return to zero after the current sheet passes the axial location. The line-integrated density is larger at the exit of the acceleration region.

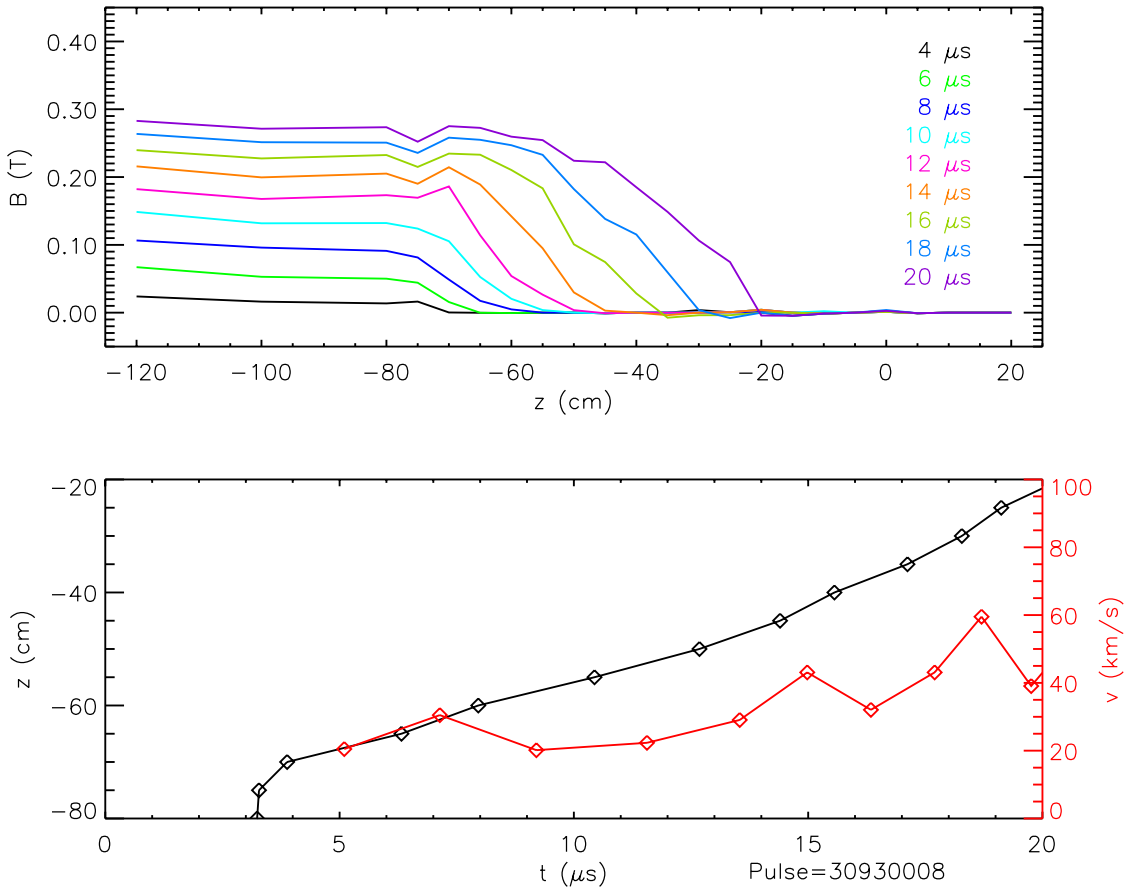

Figure 6.31: The position and velocity of the current are measured with the axial array. The magnetic fields in 2 μs intervals are shown in the top plot. The position (black) and velocity (red) of the leading edge of the current sheet along the outer electrode are shown in the lower plot. The velocity is lower than the hollow inner electrode current sheet velocities.

6.5.2 Assembly Region

After the plasma exits the acceleration region a Z-pinch is formed. The measurements of the plasma properties are made at the midplane of the Z-pinch $z = 0$ cm. Axial variations in the Z-pinch should be small at this location. A quiescent period is seen in the normalized magnetic mode activity. Photos of emission from the plasma show a stationary plasma during this time. The plasma density is peaked on the axis of the machine. Measurements of the line emission show the C III is constant during the quiescent period. A sheared flow

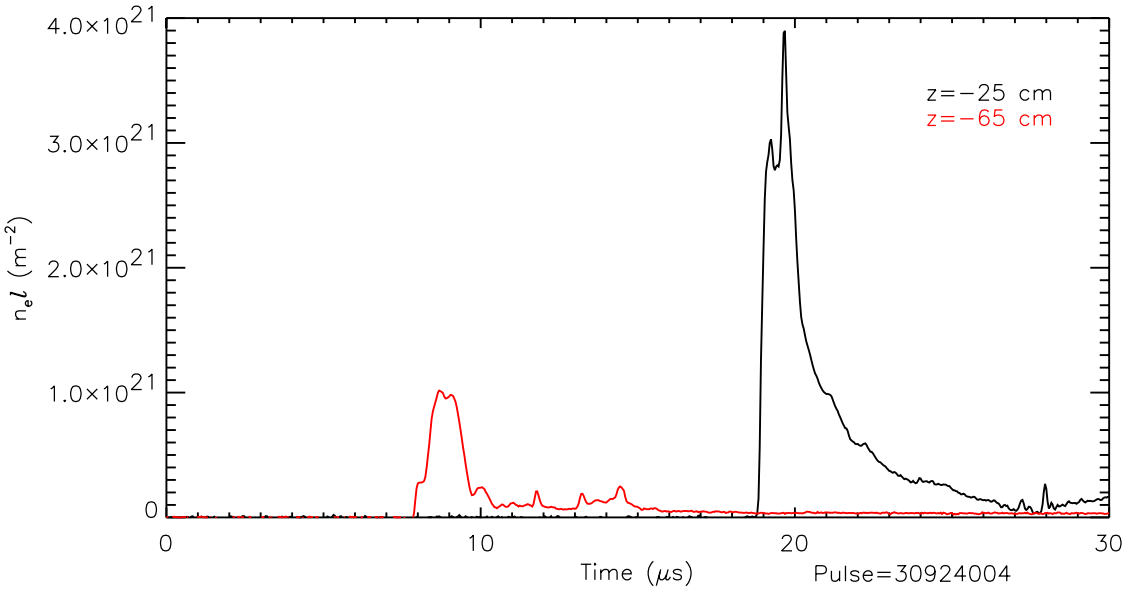

Figure 6.32: The density in the current sheet is measured with the interferometer. The density is increasing as the current sheet travels along the annulus.

is measured with the ICCD during the quiescent period.

Magnetic Field Measurements

The general character of the plasma pulse is described by the mode data. The average magnetic field and radial location of the Z-pinch is found with the azimuthal array of magnetic probes at $z = 0$ cm. The amplitudes of the Fourier components are shown in the top plot of Fig. 6.33. The average magnetic field, the m_0 component, stays at 0.1 T for 50 μs during the pulse. The normalized mode data describes the behavior of the Z-pinch. Large mode activity is measured early in time as the pinch is forming, shown in the bottom plot. Asymmetries in the current sheet cause this behavior. Shortly after formation, the normalized mode amplitudes decrease below an empirical value 0.2. The mode activity stays below this value for over 40 μs. The plasma current is located within 1 cm of the axis of the machine during this time. During the quiescent period, a Z-pinch is seen on the axis of machine. The amplitude and frequency of the normalized mode amplitudes then increases.

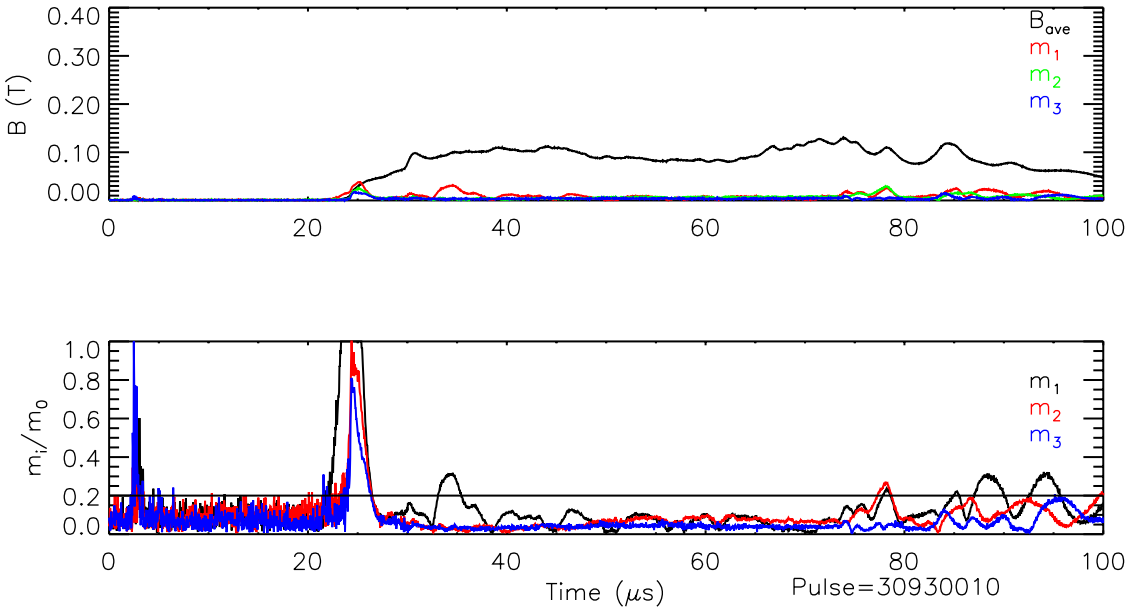

Figure 6.33: The magnetic mode amplitudes measured at the $z = 0$ cm array are shown in the top plot. The average magnetic at the wall is 0.1 T as the Z-pinch forms. The magnitude of the magnetic field increases later in the quiescent period. The normalized mode is shown in the bottom plot. The quiescent period is longer than the other studies.

At the end of the quiescent period, the amplitude of the oscillations in the magnetic mode data are seen.

Emission Measurements

The size and shape of the Z-pinch are estimated with photos of the emission from the plasma. Pictures of optical emission show a uniform Z-pinch during the quiescent period, shown in Fig. 6.34. The photos are taken through a 2 inch hole in the outer electrode at $z = 0$ cm. The end wall is on the left side of the photo. Late in time a kink can be seen in the pictures.

Radiation, emitted from the Z-pinch, is used to measure the plasma properties. The filterscopes and CCD monitor the line and bremsstrahlung radiation on each pulse. The characteristic emission from a pulse is shown in Fig. 6.35. The level of H_α is high at the

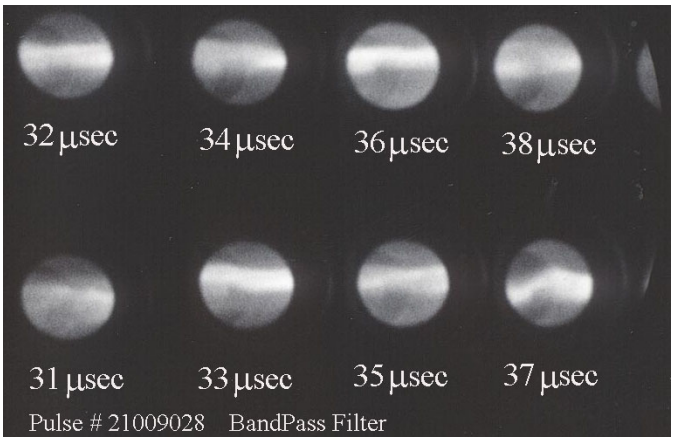

Figure 6.34: Photos of the plasma taken with the Imacon show a column of plasma during the quiescent period. A band pass filter from 500 nm to 600 nm has used for this pulse. A kink can be seen at 37 μs.

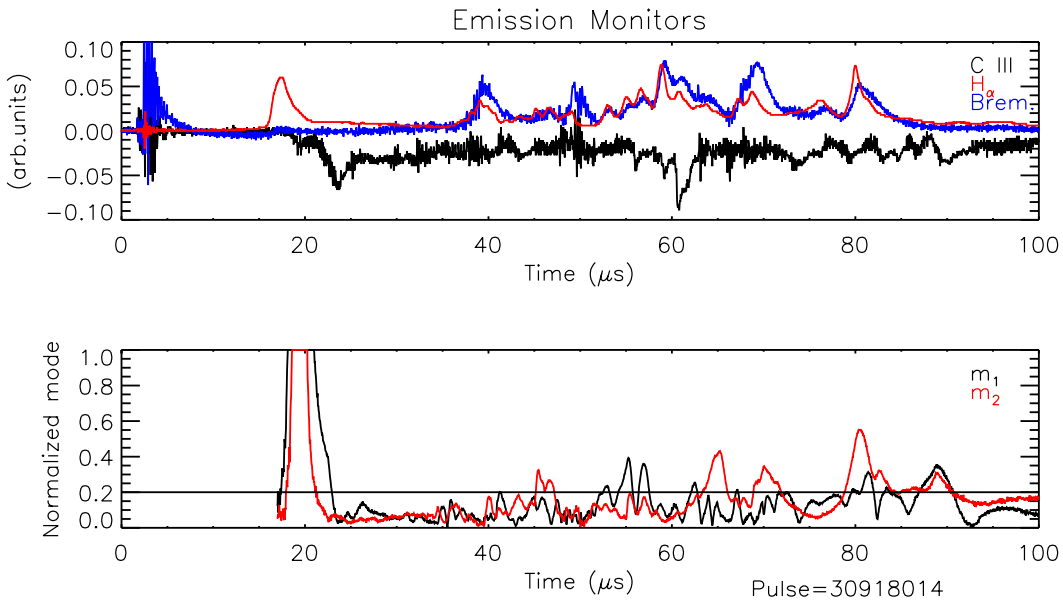

Figure 6.35: The filter scopes and PMT are used to measure the temporal evolution of the radiation. The H_α level is low during the quiescent period. The PMT is tuned to C III for this pulse. The brightness remains constant throughout the quiescent period.

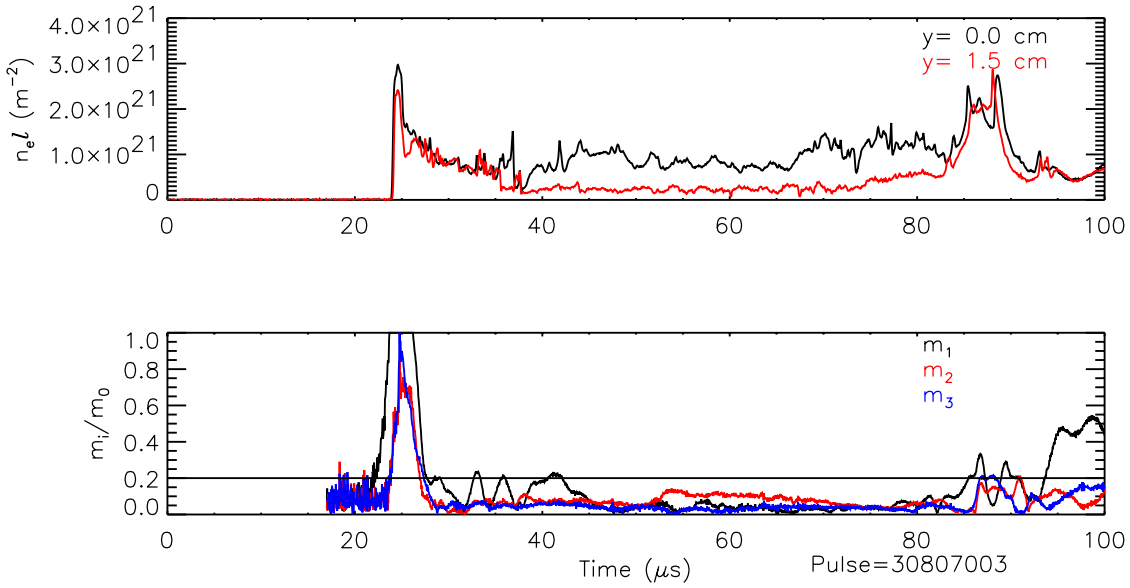

Figure 6.36: The interferometer measures the density of the Z-pinch at $z = 0$ cm. As the Z-pinch forms a uniform density is seen in the assembly region. During the quiescent period, the density is peaked on the axis of the machine. At the end of the quiescent period, a uniform density is seen.

beginning and end of the quiescent period. During the quiescent period, the amount of neutral H in the Z-pinch decreases. The bremsstrahlung radiation level is highest after the quiescent period. As instabilities form in the plasma, regions of hot, dense plasma form in the assembly region, which emit the bremsstrahlung radiation. This mechanism accounts for most of the neutron emission seen on static Z-pinches and plasma focus experiments. This is the main reason ZaP does not use D_2 as its working gas.

Density Measurements

Density measurements of the Z-pinch are made with both chords of the interferometer. The two chords are located at $z = 0$ cm. One chord is located through the center of the machine and the other chord is located 1.5 cm above the axis of the machine. The density measurements are shown in Fig. 6.36. During the formation of the Z-pinch a uniform density is measured in the machine. The density is peaked during the quiescent period.

The average density in the Z-pinch remains constant during this time. During the increased mode activity, a uniform density is seen.

Spectroscopy Measurements

The velocity profile throughout the lifetime of the Z-pinch is measured with the C III line at 229.7 nm. The intensity of the C III remains constant, shown in Fig. 6.35. This configuration also has some on the most repeatable mode characteristics. Unlike the other studies, the C III emission remains peaked in the Z-pinch. The local plasma parameters can be found during all stages of the Z-pinch.

The velocity is uniform as the Z-pinch forms. A uniform velocity is measured in Fig. 6.37. The emissivity is low, but peaked, while the initial spike in the normalized m_1 is decreasing . The velocity is 100 km/s across the Z-pinch. This velocity profile is similar to the other studies. As the activity in the modes decrease, a velocity shear begins to form, shown in Fig. 6.38. The velocity in the middle of the Z-pinch has decreased at this time. The edge velocity is still 100 km/s. The large fluctuations in the velocity from the right side deconvolution is due to the low emissivity in this region. The emissivity profile is peaked during this time.

During the quiescent period a sheared flow is measured. The ICCD data are taken in the middle of the quiescent period in Fig. 6.39 The velocity profile is similar to Fig. 6.38. The width of the emissivity profile has increased. The velocity profile evolves during the quiescent period. Figure 6.40 is towards the end of the quiescent period. The edge velocity has decreased to zero. The center velocity is still 50 km/s. The velocity shear has been reversed between these two times.

The velocity shear decreases as the mode activity increases. The velocity profile, in Fig. 6.41, was taken as the mode activity begins to increase. The edge velocity is zero at this time. The velocity in the center of the plasma has decreased to 20 km/s. The velocity profile is similar to the C IV profile in Fig. 6.27. The velocity profile in Fig. 6.27 is uniform. This profile was taken during the increased mode activity. The evolution of the velocity profile can be measured when the emission of C III across the Z-pinch is constant.

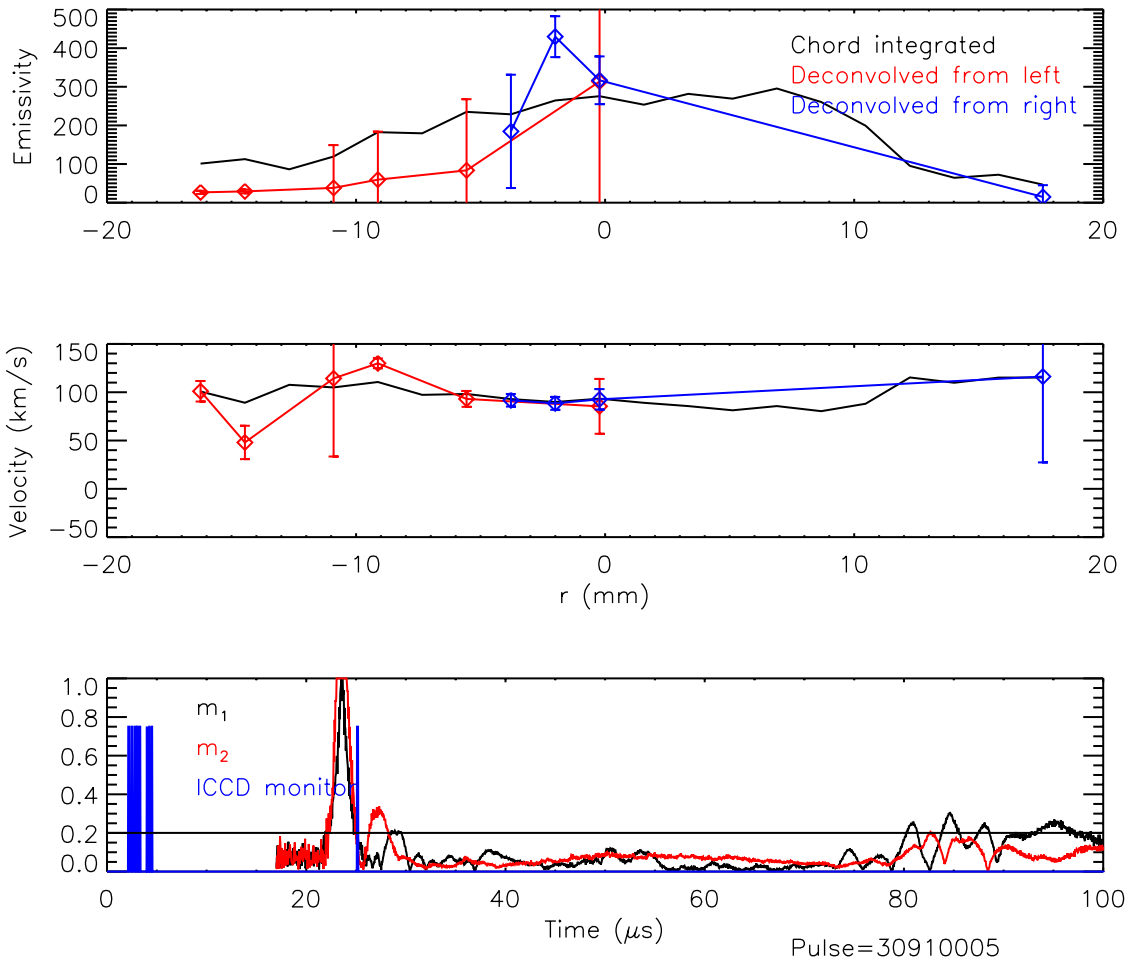

Figure 6.37: Shown are the chord integrated and deconvolved profiles of the emissivity and velocity from pulse 30910005. A uniform velocity, 100 km/s, is measured during the formation of the Z-pinch.

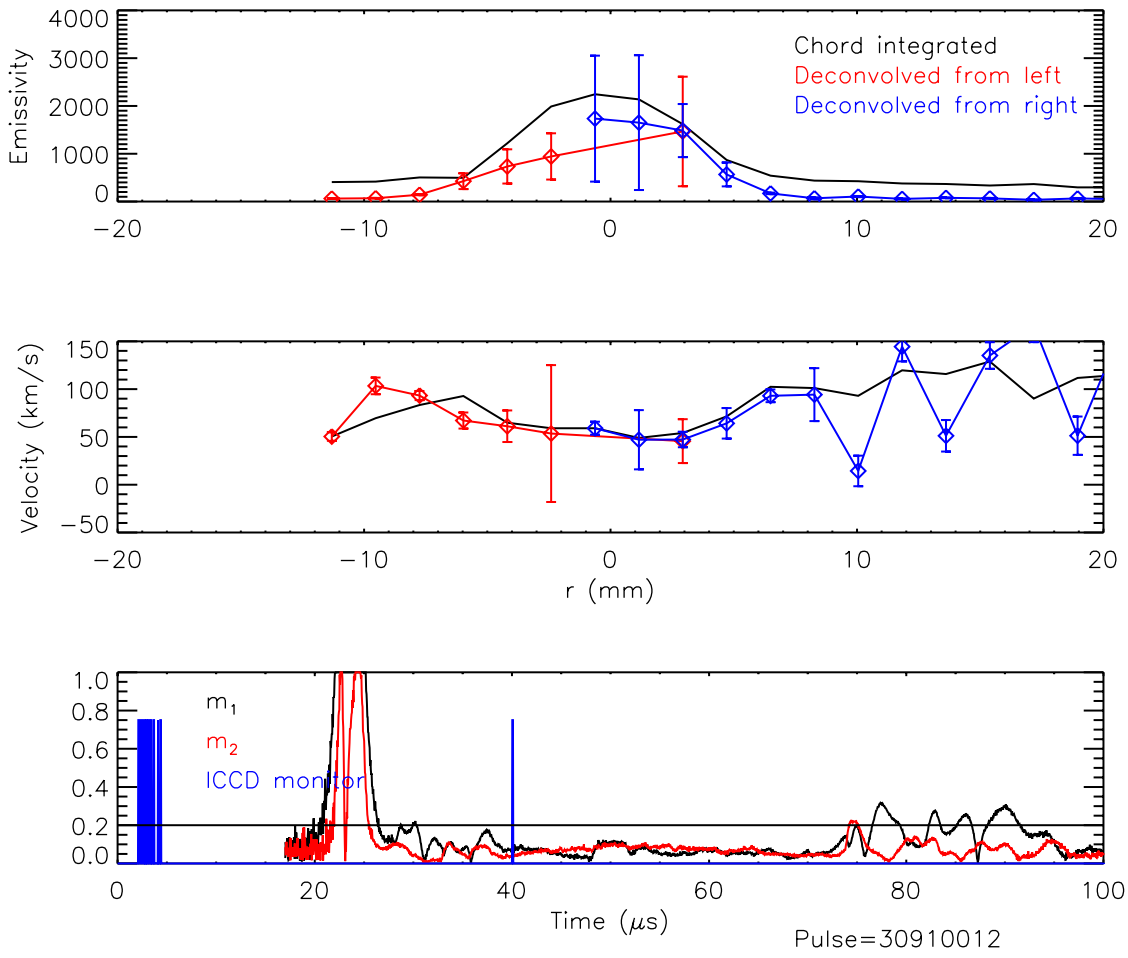

Figure 6.38: The plasma parameters measured with the ICCD show the beginning of the sheared flow in the Z-pinch. The emissivity profile is peaked on the axis of the machine during this time. The velocity shear is on the edge of the Z-pinch. The large fluctuations in the velocity from the right side deconvolution is due to the low emissivity in this region.

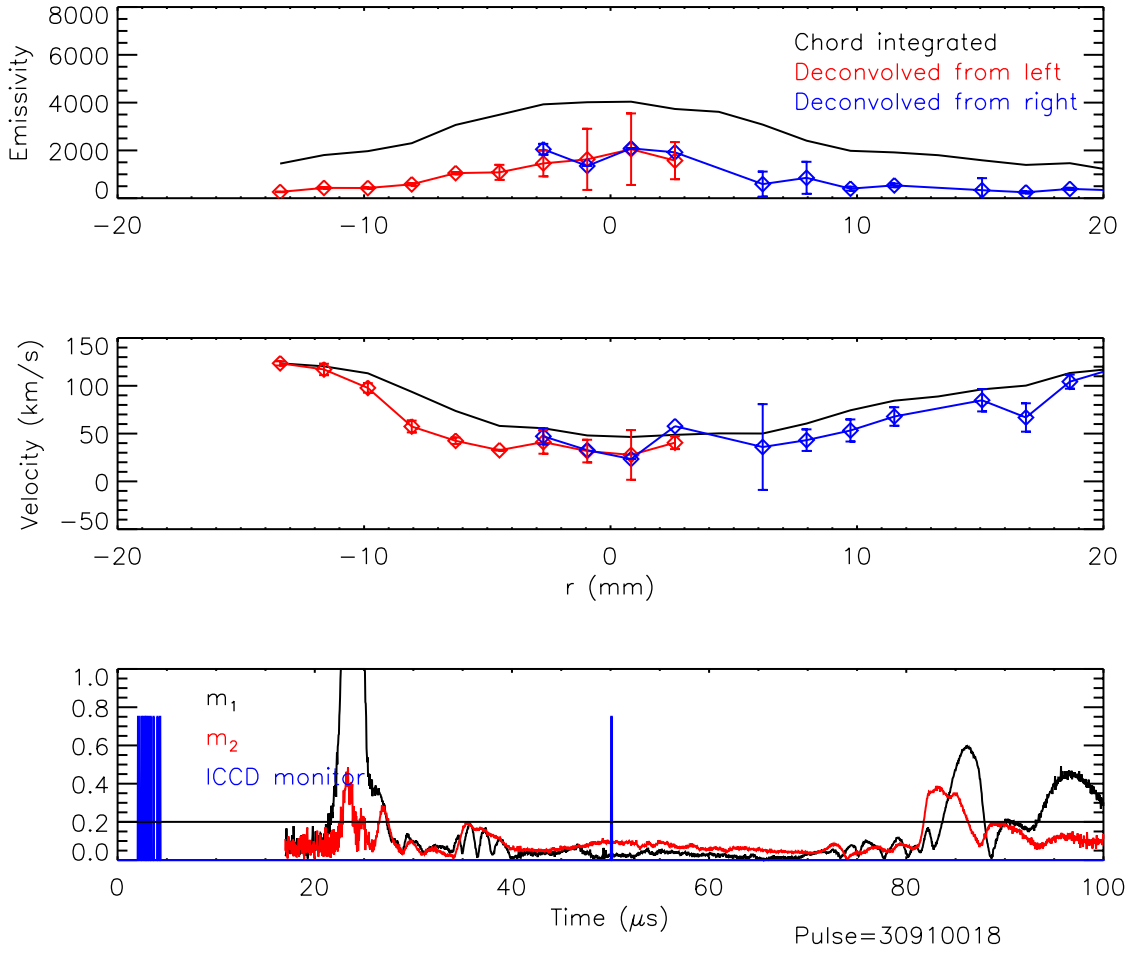

Figure 6.39: The plasma parameters measured with the ICCD show a flow profile of the Z-pinch during the middle of the quiescent period. The emissivity profile is peaked on the axis of the machine during this time. The velocity shear is on the edge of the Z-pinch.

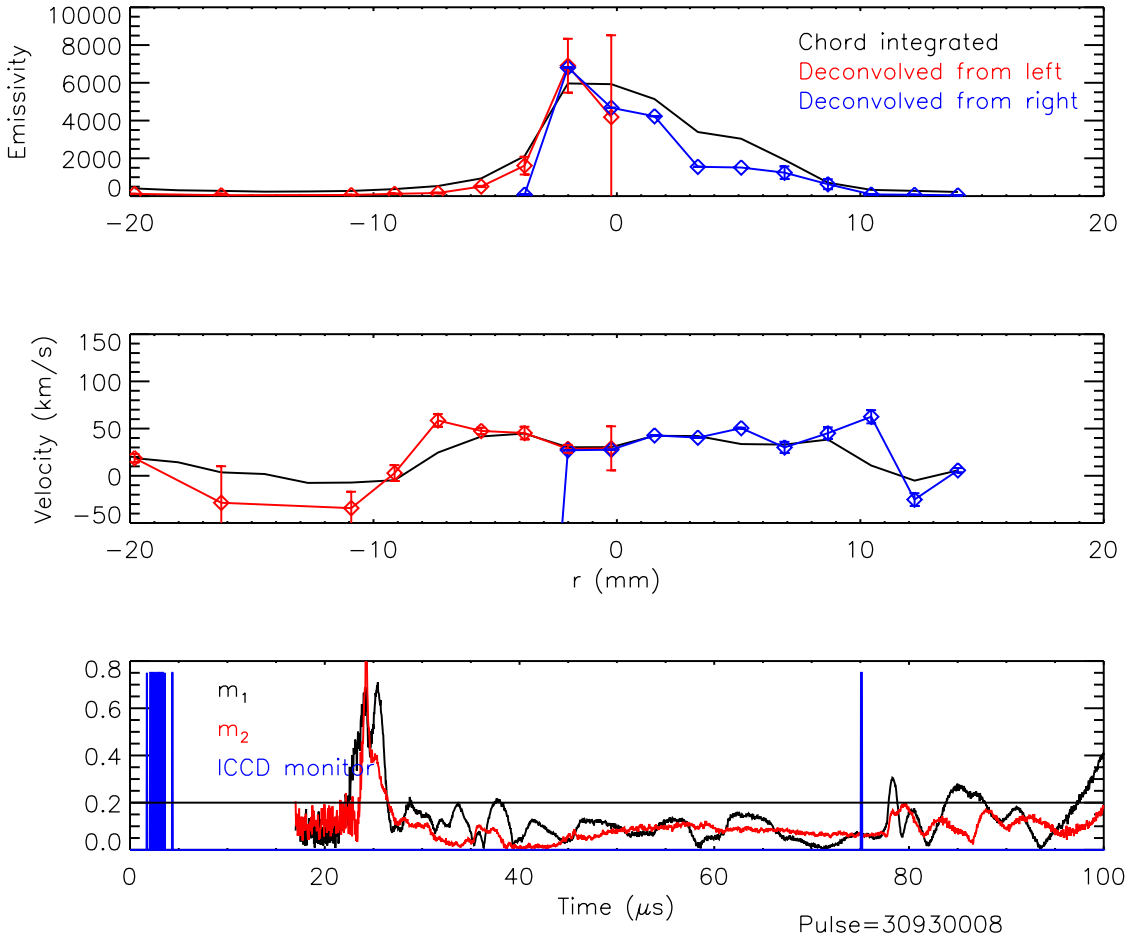

Figure 6.40: The plasma parameters measured with the ICCD show a sheared flow Z-pinch at the end of the quiescent period. The emissivity profile is peaked on the axis of the machine during this time. The velocity shear is on the edge of the Z-pinch.

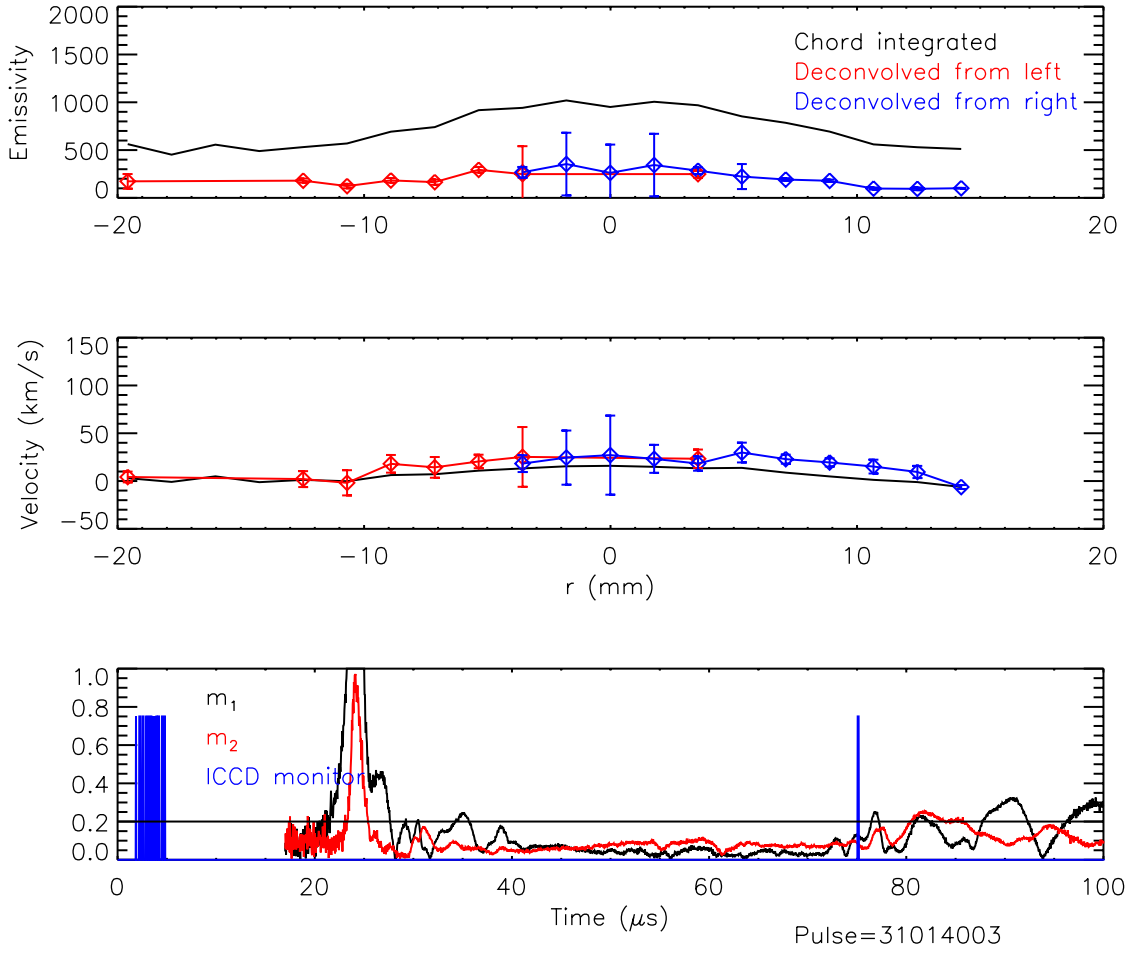

Figure 6.41: Shown is the velocity profile as the mode activity increases. The velocity shear has decreased to a low level during this time.

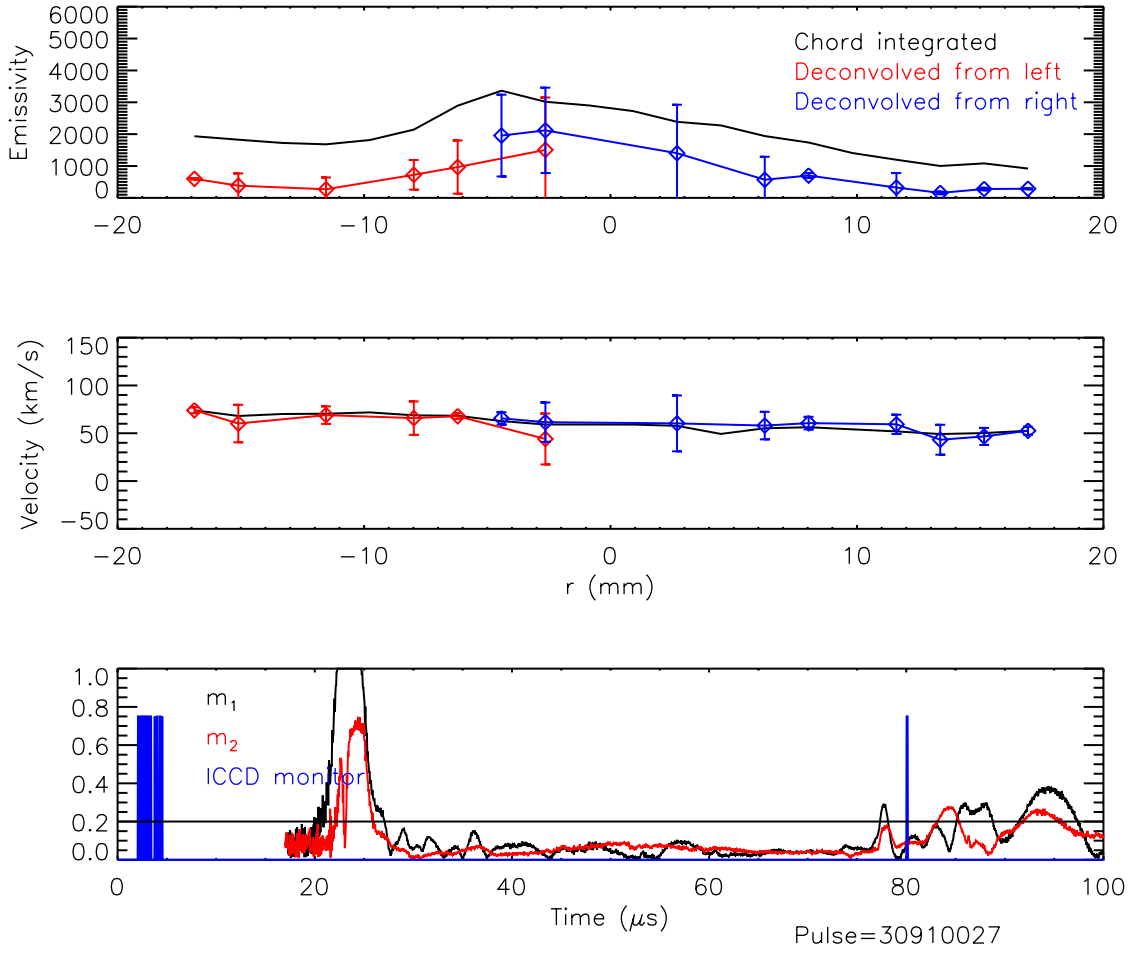

Figure 6.42: During the increased mode activity the velocity shear is low. A Z-pinch is still seen in the emission profile. The velocity is finite, but uniform.

Chapter 7

DISCUSSION OF RESULTS

Stationary Z-pinches are formed in the ZaP experiment. The breakdown of the neutral gas causes asymmetries in the current sheet. These perturbations are responsible for the initial mode activity. Multiple diagnostics have been used to show that the Z-pinch is located on the axis of the machine. All of the studies have shown that a stationary Z-pinch persists in the assembly region during the quiescent period in the mode activity. The diameter of the Z-pinch, measured for the emission photos, is about 2 cm. The density data agrees with this estimate. The walls do not stabilize the instabilities since $r_{wall}/a = 10$. As the magnetic pressure increases, higher ionization states are measured in the Z-pinch. During the quiescent period, the velocity is sheared.

As the mode activity increases, perturbations are seen in the Z-pinch. Photos of the emission from the plasma show that the perturbations measured by the magnetic field array are mostly from kinks in the Z-pinch. The data from the interferometer also suggest that kinks, not sausages, are the predominant mode. The density of the Z-pinch does not decrease during the increased mode activity. The perturbations do not cause a catastrophic failure of the total plasma current. The measured currents and voltages do not have large spikes on the waveforms. The Z-pinches formed in the ZaP experiment are similar to the "pinch-like structures which are stable under certain operating conditions" seen in the LASL experiments.

The general behavior of the plasma for all of the configurations is similar. The first section compares the experimental results to the snowplow models developed in Sec. 3.2. The formation and acceleration of the plasma behaves like the snowplow model described by other experiments. Changes in the driving circuit and gas fill have affected the speed of the current sheet. After the current sheet collapses onto the axis of the machine, a stationary Z-pinch is seen in the assembly region. The lifetime of the Z-pinch is orders of

magnitude greater than the instability growth time. Section 7.2 compares the measured flow shear to the theoretical value of $0.1kV_A$. The assumptions used to calculate k and V_A are discussed. The measured flow shear is greater that the theoretical threshold. The last section extends the description of the formation process. A heuristic model has been developed which explains how the flow in the Z-pinch is maintained.

7.1 Comparison to Snowplow Models

The behavior of the current sheet is consistent with snowplow models. The time-varying current model described in Sec. 3.2, predicts the location of the current sheet and the magnitude of the plasma current. The current sheet velocity is increasing as it travels along the acceleration region. The density in the current sheet is also increasing as neutral gas is ionized and entrained in the plasma, shown in Figs. 6.20 and 6.32. The arrival time of the current sheet, measured with the interferometer, is ahead of the magnetic field measured at the same axial location. Unlike most slug models, the snowplow model includes the shape of current sheet.

The snowplow model from Sec. 3.2.1 describes the current sheet in the ZaP experiment. The capacitance, inductance, and resistance of the capacitor bank, shown in Fig. 4.7, was verified by matching the current from a calibration pulse to an RLC circuit. The fill density profile of the accelerator is unknown. The total amount of gas injected into the annulus can be estimated from the characterizations described in Sec. 4.2. The gas was assumed to uniformly fill the accelerator region for this model. The plowing efficiency, k_i, was set to one and γ was equal to 1.4 for the simulations. The results from the time dependent model are compared to pulse 20718030, shown in Fig. 7.1. The model predicts the magnitude of the current and location of the current sheet. Varying the fill density changes the velocity of the current sheet. The current is not affected until the current sheet exits the acceleration region.

The behavior of the current and current sheet velocity agree with trends seen in the model. The methane and IDS studies used the same capacitor bank with a higher fill density in the methane studies. The currents for both studies were the same. Changing the

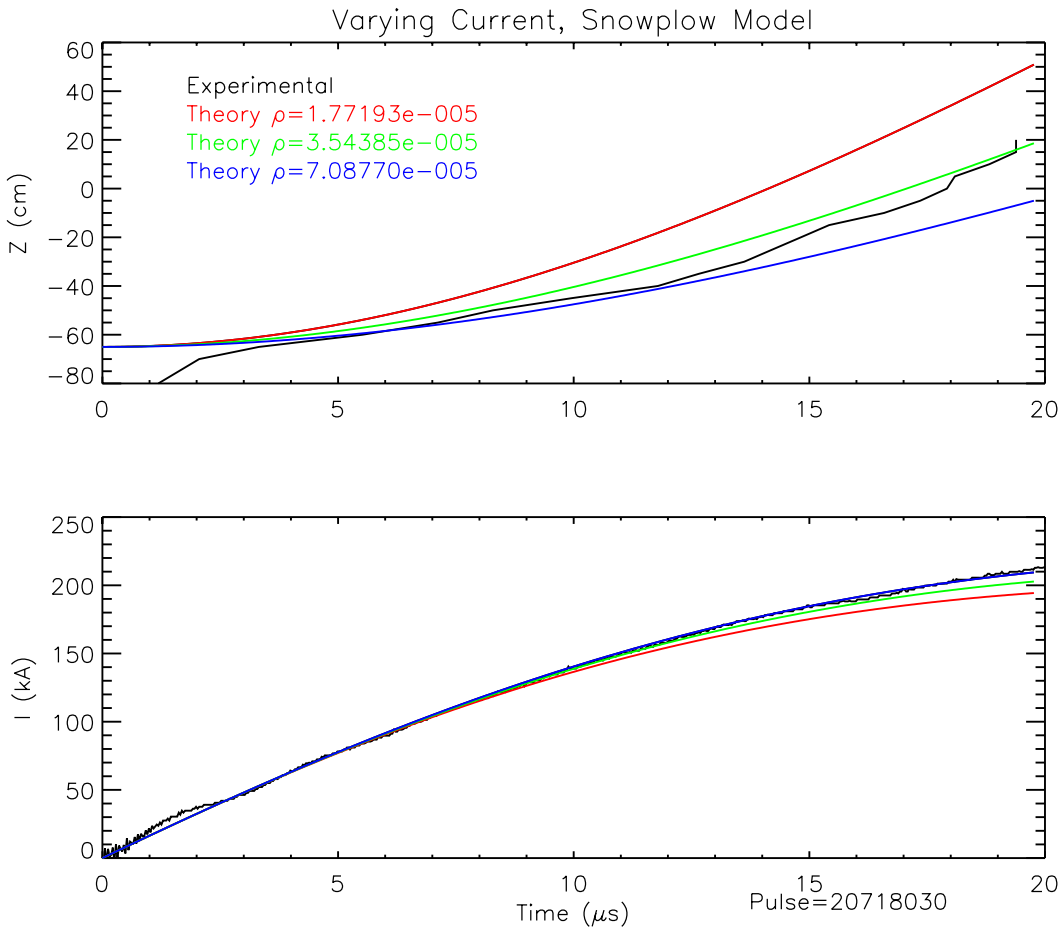

Figure 7.1: The experimental position and current (black) agrees with the theoretical prediction (colored lines). The three fill densities are consistent with the characterization. Changing the fill density changes the velocity without significantly affecting the current.

speed of the current sheet by changing bank voltage does not affect the current. The initial current waveforms in Fig. 4.10 are the same for different bank voltages. As the density increases, the acceleration of the current sheet is reduced. The methane studies had a lower velocity of the current sheet than the IDS studies. The current sheet shape, used in model, is not consistent with the shape measured by the LASL group. The position and currents agree, because the difference in the inductance of the two shapes is small and the inductance of the capacitor bank is much larger than the annulus.

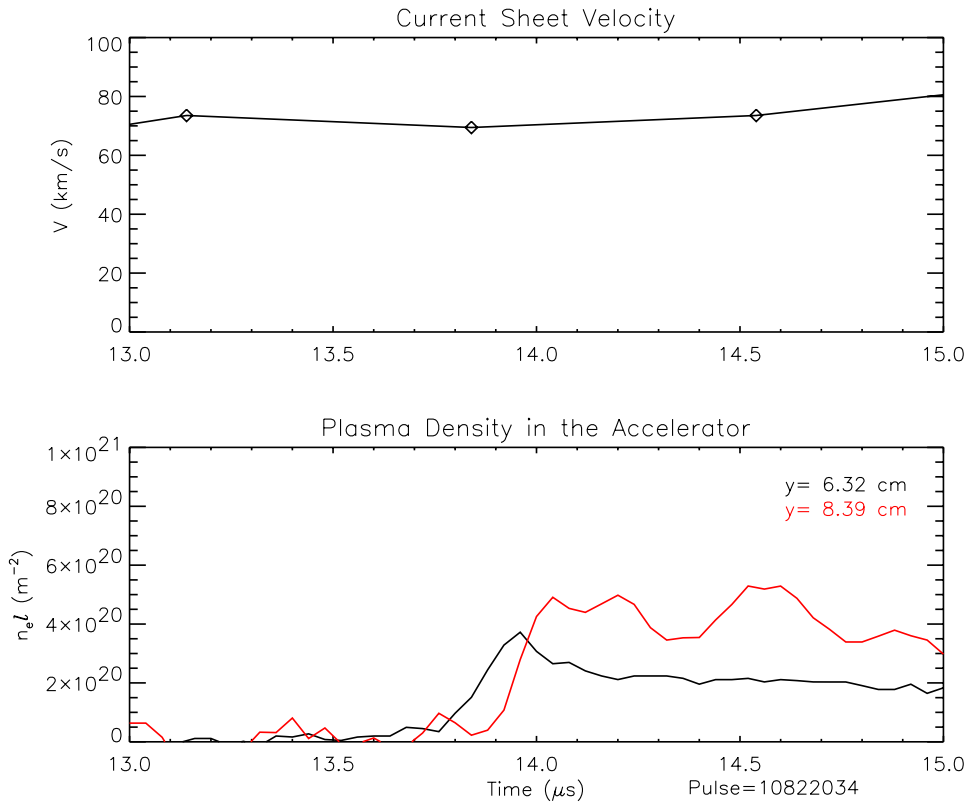

Figure 7.2: Shown are velocity of the current sheet and interferometry data from two impact parameters taken at z=-25 cm. The two sets of data are used to find the slope of the current sheet.

The exact shape of the current sheet cannot be measured on the ZaP experiment. The current sheet shape, shown in Fig. 3.4, does not change by a significant amount when the initial conditions change. The slope of the current sheet can be found by looking at the time difference between the two chords of interferometry and calculating the velocity with the axial magnetic probe array, shown in Fig. 7.2. The velocity of the current sheet is 75 km/s as it passes the axial location of the interferometry chords. The difference of the impact parameters of the two chords is 2.02 cm, and the upper chord begins to measure a density 0.1 μs after the lower chord. The measured slope of 2.6 is steeper than the calculated slope of LASL model. A radially decreasing distribution of neutral gas in the accelerator could also explain the discrepancy between the current sheet shapes. The model assumes a constant

current in the snowplow. It was verified, by removing the outer electrode and letting the current sheet expand into a vacuum. In the ZaP experiment, the plasma is compressed against the outer electrode.

7.2 Comparison to the Sheared Flow Theory

A sheared flow Z-pinch is formed in the ZaP experiment. While, many of the plasma properties have been measured with the avaliable diagnostics, the pressure profile has not. Assumptions can be made to calculate V_A, the kink growth time, and the flow shear required to stabilize kinks. The measured plasma parameters can then be compared to the theoretical predictions.

7.2.1 Calculation of the Z-pinch Equilibrium and Stability

Presently, the equilibrium pressure profile has not been measured on the ZaP experiment. The avaliable measurements are consistent with a stable, hot, dense Z-pinch in the center of the machine. Since the Z-pinch can be made stable to m_0 perturbations by controlling the pressure profile and most of the perturbations seen in the Z-pinch are due to m_1 activity, the pressure profile is assumed to satisfy Eq. 1.8. Future work on the ZaP experiment will include efforts to measure the pressure profile.

Verification of Eq. 1.9 requires the flow profile, k, and V_A must be measured. The flow profile has been measured for various experimental configurations. The Helium study showed that the velocities measured with passive spectroscopy are similar to the bulk plasma. Measurements of Alfvén velocities and k are not possible at this time. Estimates of these two quantities can be made with measurements of the pinch radius, the magnetic field at the wall, and the average density in the Z-pinch. The diameter of the Z-pinch, d, is measured with the Imacon photos. The interferometry and ICCD data agree with this estimate. The shortest wavelength, given by $\lambda = d/2\pi$, places the largest constraint on the flow shear. By combining the pinch radius, the magnetic field at the wall, and assuming all the current is flowing in the Z-pinch, the magnetic field at the edge of the pinch, B_a, is known. This assumption can be verified by measuring the temperature and comparing the

maximum plasma pressure, P_0, and magnetic pressure

$$P_0 = \frac{B_a^2}{\mu_0}, \tag{7.1}$$

if the edge pressure is zero. If impurities in the Z-pinch are neglected, $n_i = n_e$, the average density in the Z-pinch, $\bar{\rho}$, is given by

$$\bar{\rho} = m_i \bar{n}_e \tag{7.2}$$

where m_i is the mass of a proton and $\bar{n}_e$ is the average number density in the Z-pinch. The average density of the Z-pinch and the magnetic field at the edge of the Z-pinch are then used to define the Alfvén velocity

$$V_A = \frac{B_a}{\sqrt{\mu_0 \bar{\rho}}} \tag{7.3}$$

The measured velocity shear is then compared to $0.1kV_A$.

7.2.2 Experimentally Measured Shears Agree with the Theory

The experiment is studying sheared flow stabilization of plasma instabilities. The velocity shear has been correlated with the quiescent period. The mode activity is large when the shear is small. The assumptions made in Sec. 7.2.1 are used to calculate the parameters of the Z-pinch. The results from the hollow inner electrode and nose cone studies are summarized in Table 7.1. The measured plasma parameters for the hollow inner electrode study are averaged over the time interval which the ICCD data was taken on pulse 726025. The average plasma parameters from the quiescent period are used for the nose cone study. The two configurations have similar characteristics in spite of the different electrode configurations and capacitor banks.

The amount of velocity shear required to stabilize the ideal MHD modes is different for various authors. The results during the quiescent period had large, local velocity shears. The velocity shear from pulse 31014003, shown in Fig. 7.3, shows that the threshold may be $0.1kV_A$. The velocity shear is shown on the second graph. The shear is lower than $0.1kV_A$ as the mode activity increases. The chord-averaged velocity profile is similar in Fig. 6.27. Using the formation process described in Sec. 3.1, the ZaP experiment has generated Z-pinches which show static MHD instabilities are suppressed with sheared flows.

Table 7.1: The plasma parameters from two of the studies are similar. Equation 7.2 is not valid for the methane studies. In spite of the different bank energies used, the Alfvén velocity and temperature of the Z-pinch are the same. dV_z/dr is calculated using the velocity in the center of the Z-pinch and assuming the velocity linearly decreases to zero on the edge. The measured velocity shears are larger on the edge.

Parameter		Hollow Inner Electrode Study	Nose Cone Study
a	(cm)	1	1
B_{wall}	(T)	0.124	0.1
B_{a}	(T)	1.24	1.0
n_e	(m^{-3})	3.2x10^{22}	2x10^{22}
$(T_i + T_e)_{\text{calculated}}$	(eV)	240	250
$(T_i)_{\text{measured}}$	(eV)	100-150	
V_A	(km/s)	150	150
$0.1kV_A$	(s^{-1})	4.7x10^6	4.7x10^6
dV_z/dr	(s^{-1})	$\geq$1.0x10^7	$\geq$6.0x10^6
τ_{growth}	(ns)	21	21
$\text{t}_{\text{quiescent}}$	(μs)	15-20	30-40

7.3 Deflagration Process

The formation process generates a Z-pinch with an embedded flow. The sheared velocity in the Z-pinch has been correlated with the quiescent period in the magnetic modes. Another process in the accelerator is ejecting plasma during the quiescent period. The flow in the Z-pinch may be maintained by the deflagration process describe by D.Y. Cheng [14]. The formation and sustainment process can be explained with a heuristic model. The model and the experiment measurements leading to this description are presented in this section.

During the quiescent period, a source of plasma is needed at the end of the inner electrode to drive the plasma velocity. The inner region of the Z-pinch is flowing during the quiescent period. A sheared flow requires a source and sink of plasma at the ends of the Z-pinch to

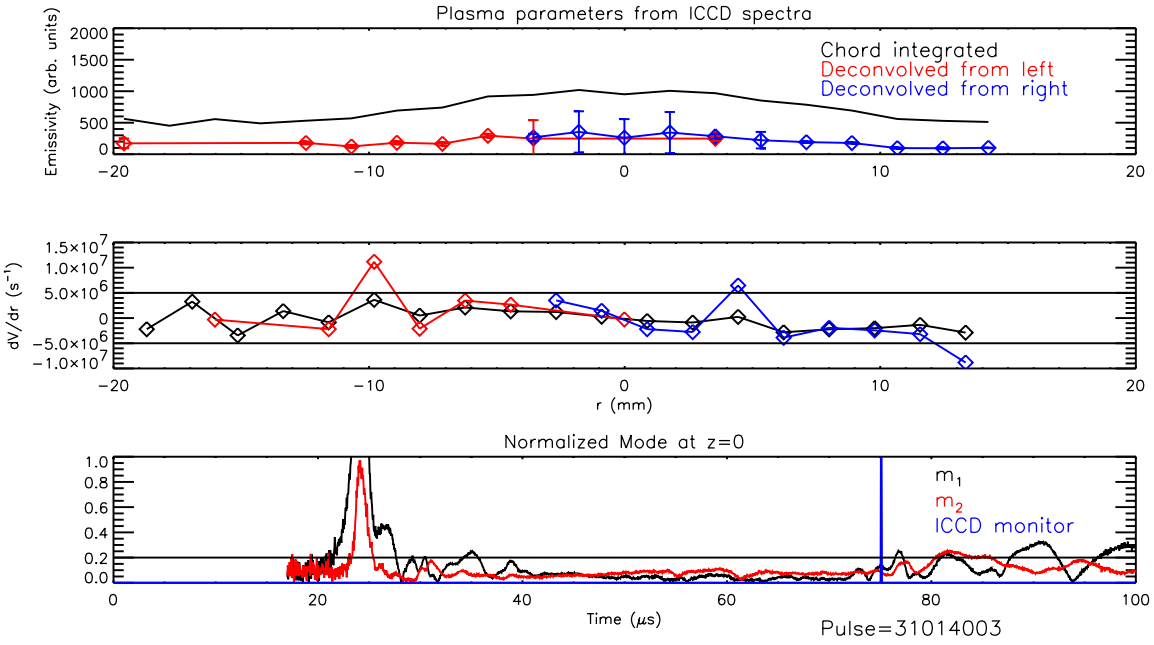

Figure 7.3: The velocity shear is below the threshold as the mode activity increases. The horizontal lines on the second graph are at $\pm 0.1 kV_A$. The velocity shear is below the threshold as the mode activity increases.

satisfy the continuity equation. There are two possible sources of plasma in the accelerator, the slow plasma or the accelerator operating in a deflagration mode. A deflagration mode in the accelerator injects plasma into the acceleration region at the end of the inner electrode. The plasma is then entrained in the Z-pinch driving the velocity profile. Although other sources of plasma may exist, the deflagration mode appears to be the driving mechanism for the velocity shear.

7.3.1 Initial Breakdown

The initial formation process is the same as was described in Sec. 3.1. The breakdown of the neutral gas is shown in Fig. 7.4. The initial gas fill in the annulus is shown in the top figure. Changing the gas puff timings adjusts the initial density profile in the annulus. The voltage is applied, breaking down the gas, and forming a uniform current sheet. The snowplow model describes the acceleration of the current. The neutral gas in front of the current

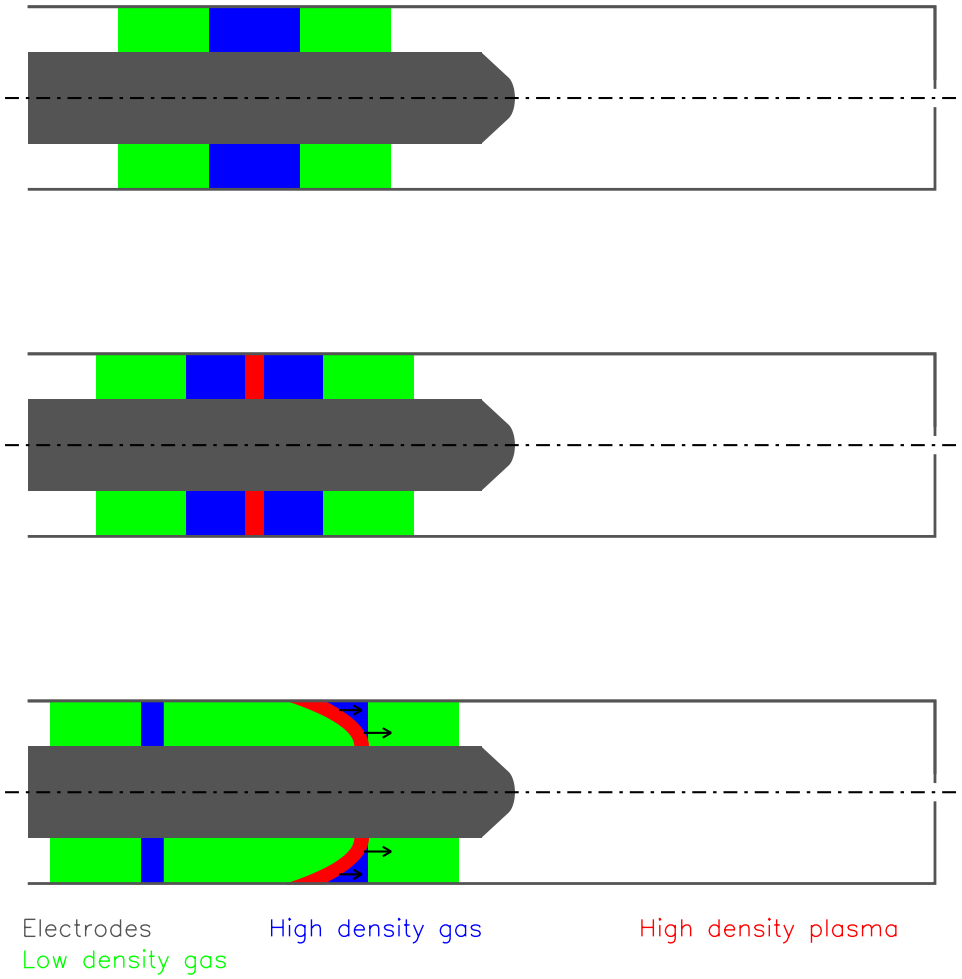

Figure 7.4: Shown is a heuristic picture of the initial steps in forming a sheared flow Z-pinch. The dashed line is the center line of the experiment. The velocity of the plasma is shown as black arrows. The rest of the colors are described at the bottom of the frames. The first frame shows the distribution of neutral gas just before the voltage is applied. The second frame shows the breakdown of the gas after the voltage is applied. The initial rundown of the current sheet is shown in the last frame.

v

sheet is ionized in the shock and entrained in the current sheet. Plasma is also pushed up against the outer electrode by the snowplow. Neutral gas is left behind the snowplow.

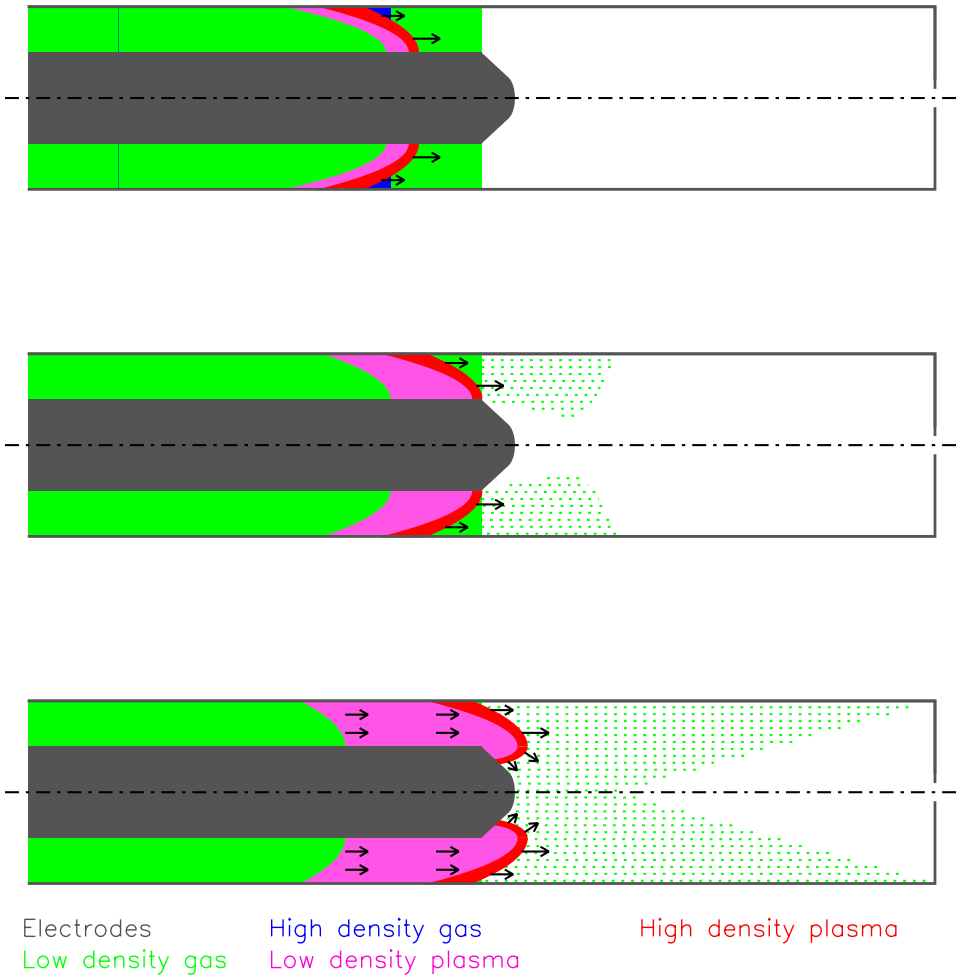

Figure 7.5: Shown is a heuristic picture of the current sheet as it reaches the end of the inner electrode and begins to form a sheared flow Z-pinch. The dashed line is the center line of the experiment. The velocity of the plasma is shown as black arrows. The rest of the colors are described at the bottom of the frames. The frames show the plasma (current distribution) beginning to widen along the outer electrode.

7.3.2 Transition During Rundown

The current sheet widens as it travels towards the acceleration region. As the current sheet travels towards the assembly region, the trailing edge begins to move in the $-z$ direction, shown in Fig 7.5. The widening of the current sheet is a transition from a snowplow to a deflagration process during the rundown [69]. The amplitudes of the normalized mode at

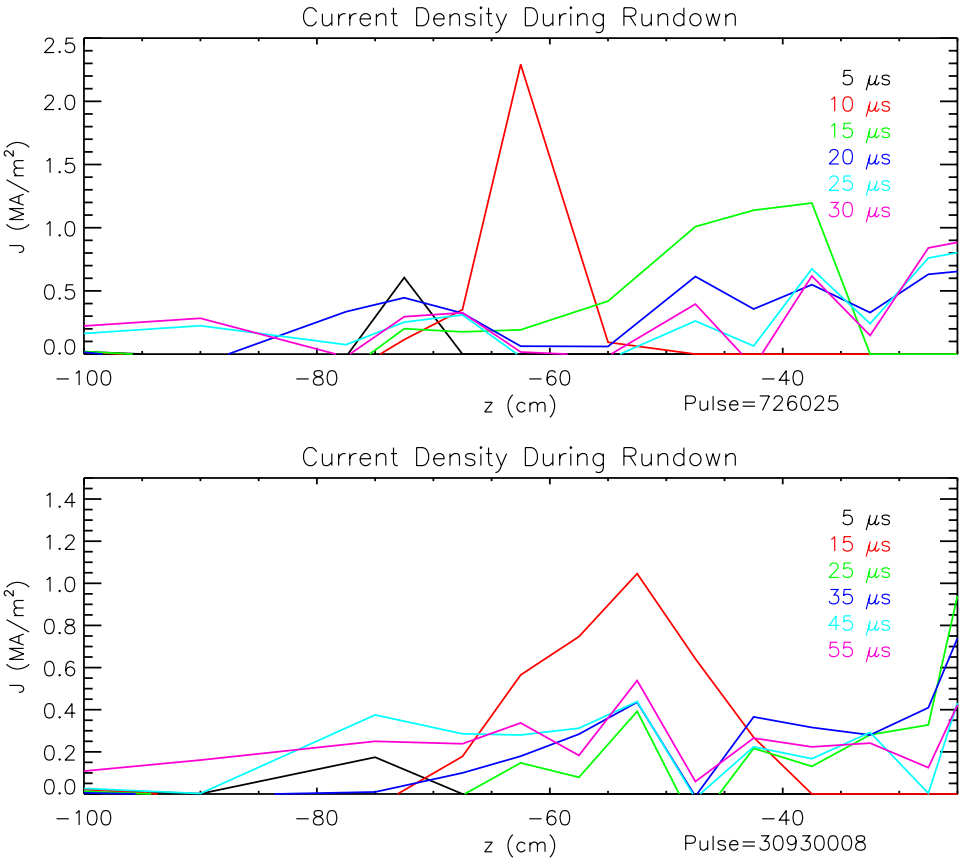

Figure 7.6: Shown are the current densities along the outer electrode at 5 μs and 10 μs intervals along the outer electrode for the two configurations. The current sheet is wider than the snowplow model prediction. The current density moves in the minus z direction.

$z = -25$ are small. The slope of the magnetic field along the outer electrode decreases as the current sheet is traveling towards the assembly region. The radial current density along the outer electrode is given by Eq. 5.26. Shown in Fig. 7.6 are the current densities along the length of the accelerator. The first two times show a well defined current sheet. As the current sheet travels along the accelerator, it begins to widen. The trailing edge remains near the gas injection ports and then begins moving in the opposite direction of the $\mathbf{J} \times \mathbf{B}$ force on the plasma. The current attachment along the outer electrode will only move in the positive z direction if the slow plasma is the source. The attachment point may move in the $-z$ direction when the accelerator operates in a deflagration mode. This behavior is

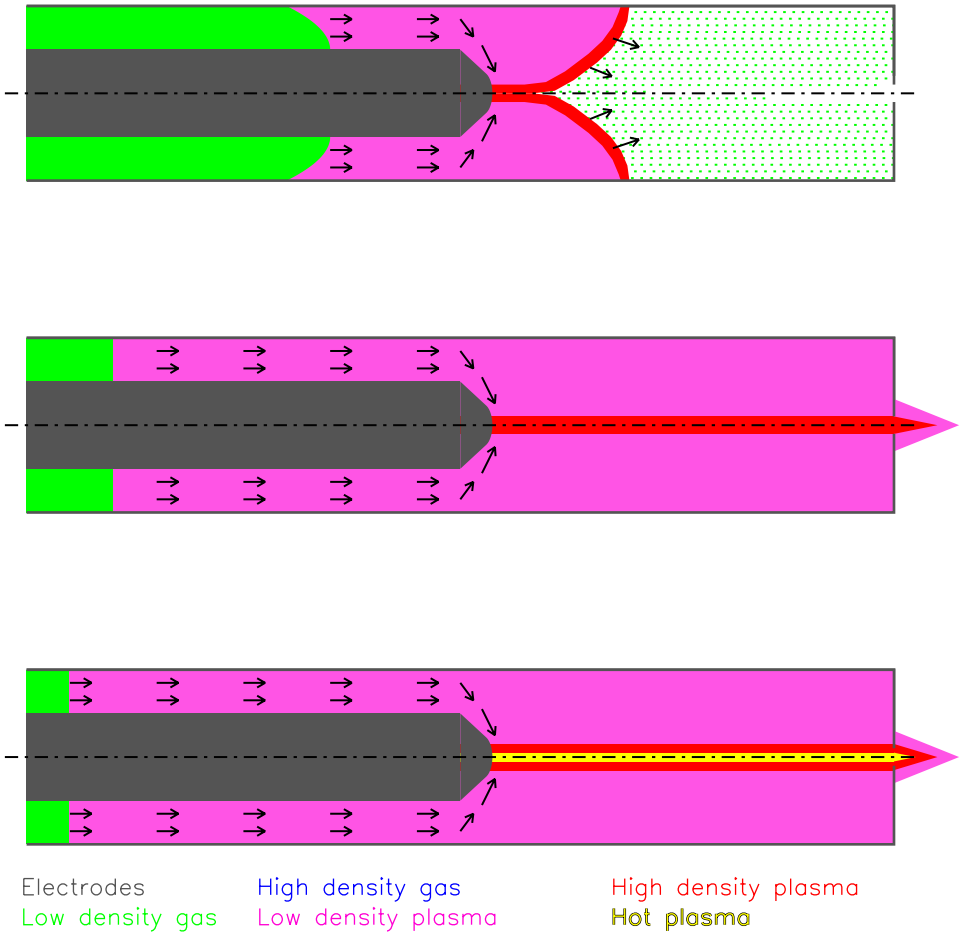

Figure 7.7: Shown is a heuristic picture of the final phases of the formation of a sheared flow Z-pinch. The dashed line is the center line of the experiment. The velocity of the plasma is shown as black arrows. The rest of the colors are described at the bottom of the frames. The first frames show the zippering action of current sheet as it collapses forming the Z-pinch. The last two frames show the final configuration of the Z-pinch. Plasma from the accelerator is consistently being supplied to the Z-pinch.

indicative of a deflagration mode in the accelerator.

7.3.3 The Flow is Maintained during Quiescent Period

Once the current sheet reaches the end of the inner electrode the entire current sheet collapses onto the axis. For this mode to exist plasma must be present in the acceleration

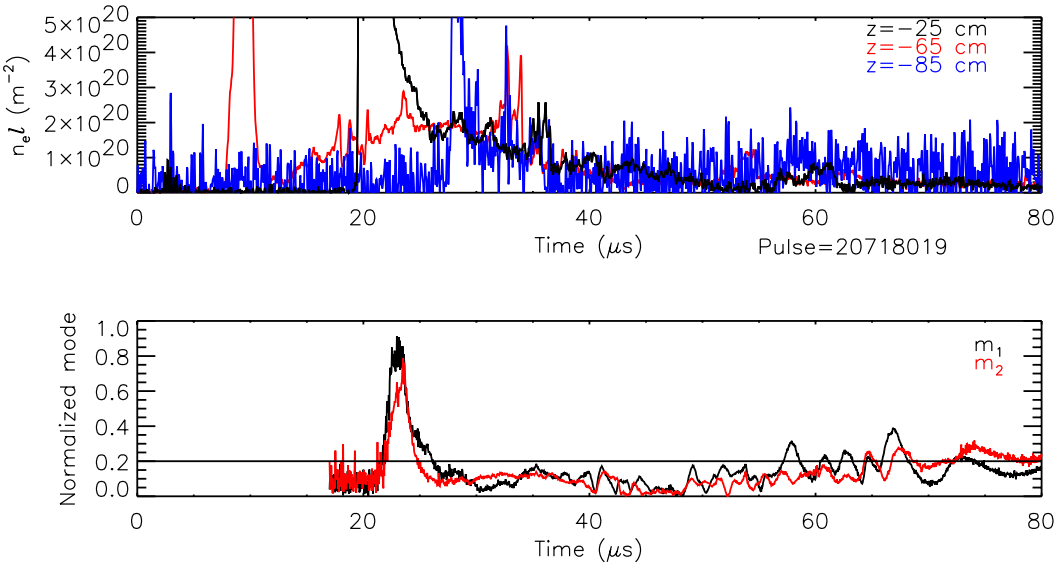

Figure 7.8: Plasma is measured upstream from the breakdown, during the quiescent period. Shown are the line integrated densities for $z = -25$ cm (black), $z = -65$ cm (red), and $z = -85$ cm (blue). During the quiescent period plasma is present at all locations. As the plasma density decreases at $z = -25$ cm the mode activity increases.

region. This mode has current flowing in the acceleration region. The current cannot flow if plasma is not present. Plasma is present in the accelerator, shown in Figs. 7.8 and 7.9. The large, steep change in the density indicates that the snowplow passing that location. The density then decreases to approximately zero, consistent with the snowplow model. As the density increases, the mode activity at $z = 0$ cm decreases. The density stays elevated throughout the quiescent period. As plasma density decreases the mode activity increases.

7.3.4 End of Quiescence is Correlated the End of the Deflagration

The quiescent period ends, when either the source or the sink at the ends of Z-pinch is not able to maintain the flow in the Z-pinch, see Fig. 7.10. The plasma flow in the Z-pinch is maintained by the deflagration process in the acceleration region. When the process ends, instabilities are seen in the Z-pinch.

The plasma is being accelerated out of the acceleration region during the quiescent

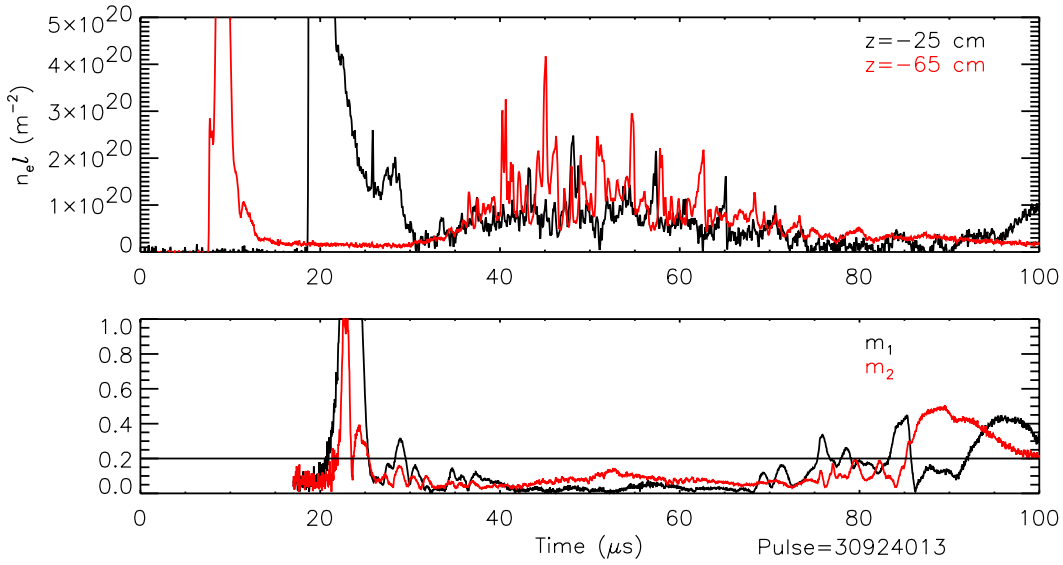

Figure 7.9: During the quiescent period plasma is present in the accelerator. Shown are the line integrated densities for $z = -25$ cm (black) and $z = -65$ cm (red) from configuration 2. During the quiescent period plasma is present at both locations. As the plasma density decreases the mode activity increases.

period. There is a $\mathbf{J} \times \mathbf{B}$ force on the plasma as shown in Fig. 7.11. The equations used to generate these contours have been explained in Sec. 5.5.5. The partial of the contours with z is proportional to the current density at the outer electrode normalized by the total current in the acceleration region. The parabolic shape of the 10% contour from 0 to 20 μs shows the leading edge of the current traveling along the outer electrode during the formation process. The close proximity of all the contours show a narrow current sheet at this time. As the current sheet travels along the outer electrode the current sheet widens as was shown in Fig. 7.11. The 90% contour's change in shape at 10 μs is due to the current sheet widening. As the current sheet continues to widen, more contours deviate from the 10% contour. The location of the current attachment points in the accelerator can be monitored with this graph.

In the assembly region, the contours have a different interpretation. The assumptions used to derive the current density calculation are not valid. All of the current in this region

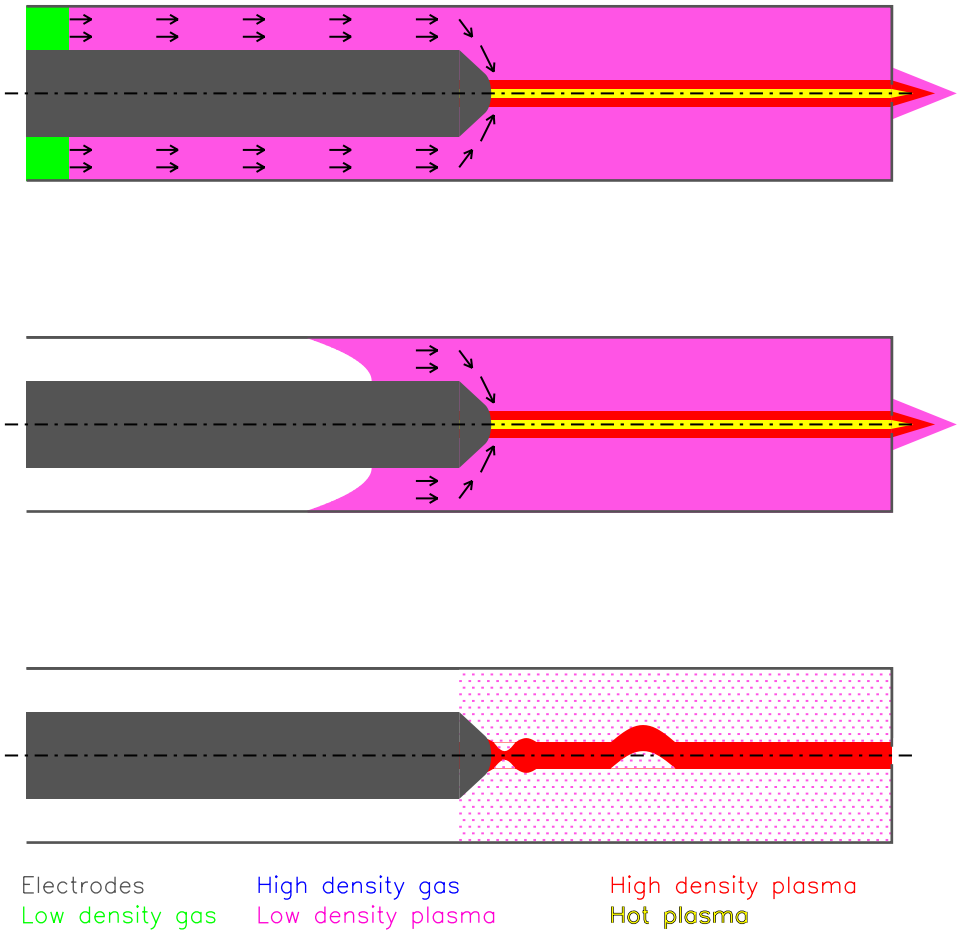

Figure 7.10: Shown is a heuristic picture of the final phases of the sheared flow Z-pinch. The dashed line is the center line of the experiment. The velocity of the plasma is shown as black arrows. The rest of the colors are described at the bottom of the frames. The first three frames show the a deflagration mode supplying plasma to the Z-pinch. During this time plasma is piling up on the end wall. The last frame shows the plasma going unstable as the deflagration mode ends.

is assumed to be in the Z-pinch. The perturbations in the magnetic field are due to the motion of the Z-pinch off the axis of the machine. The change in the magnetic field due to displacements of the Z-pinch is explained in Sec. 5.5.4. Perturbations in the plasma toward the axial array will increase the normalized magnetic field. The growth and motion of the perturbations can be seen in the 20% contours from 40 to 45 μs, see Fig. 7.12. The entire

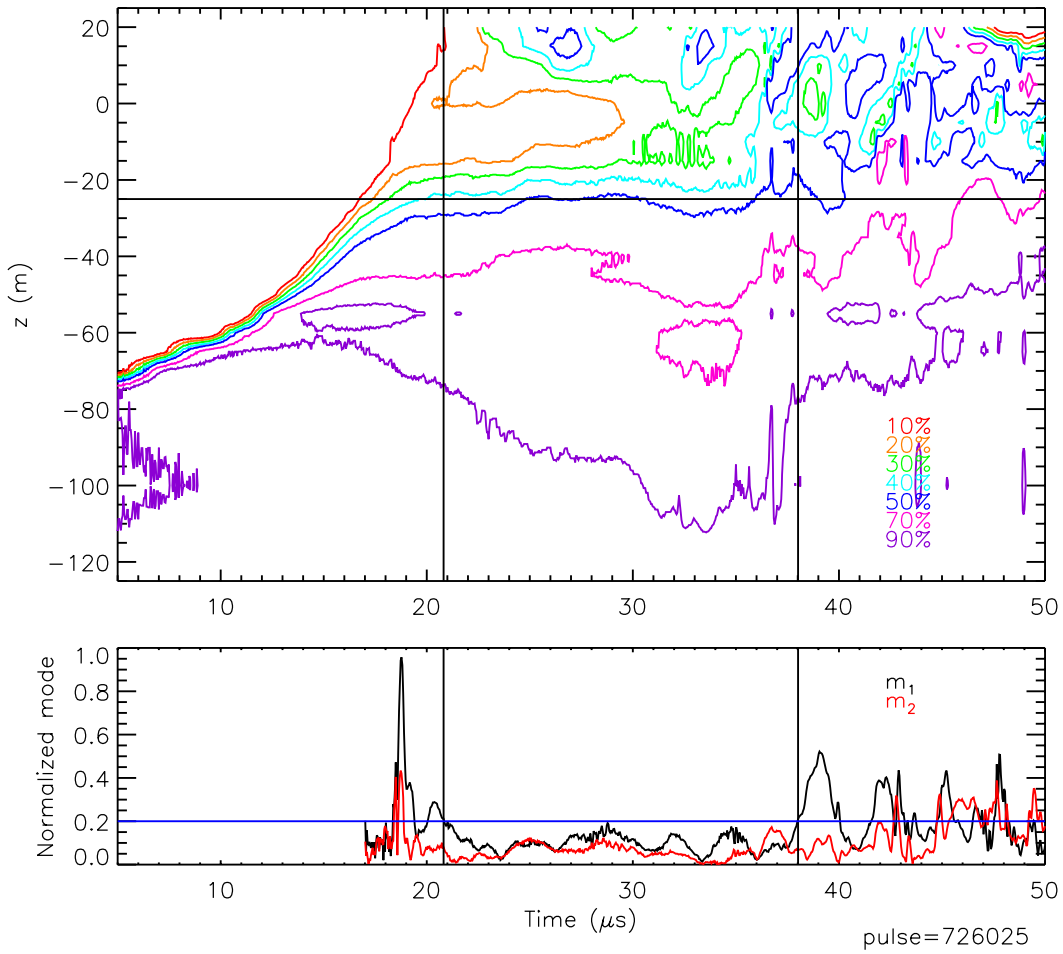

Figure 7.11: Magnetic data normalized by the current showing the axial dependence of the current. The horizontal line is at the end of the acceleration region. Below this line the contours represent axial current attachment locations. Above the line the contours are used to follow perturbations ion the Z-pinch. The horizontal lines indicate the beginning and end of the quiescent period. The perturbations, the green contour at 35 μs, in the Z-pinch are moving away from the acceleration region.

behavior of the plasma during a pulse can be understood with these two plots.

The quiescent period ends when the current density in the accelerator goes to zero. During the quiescent period half of the total current is connected inside the acceleration region. This implies there is a $\mathbf{J} \times \mathbf{B}$ force on the plasma in the accelerator. The ionization front is between the 50% and 90% lines. The oscillations seen in Fig. 7.12 from 50 to 70 μs

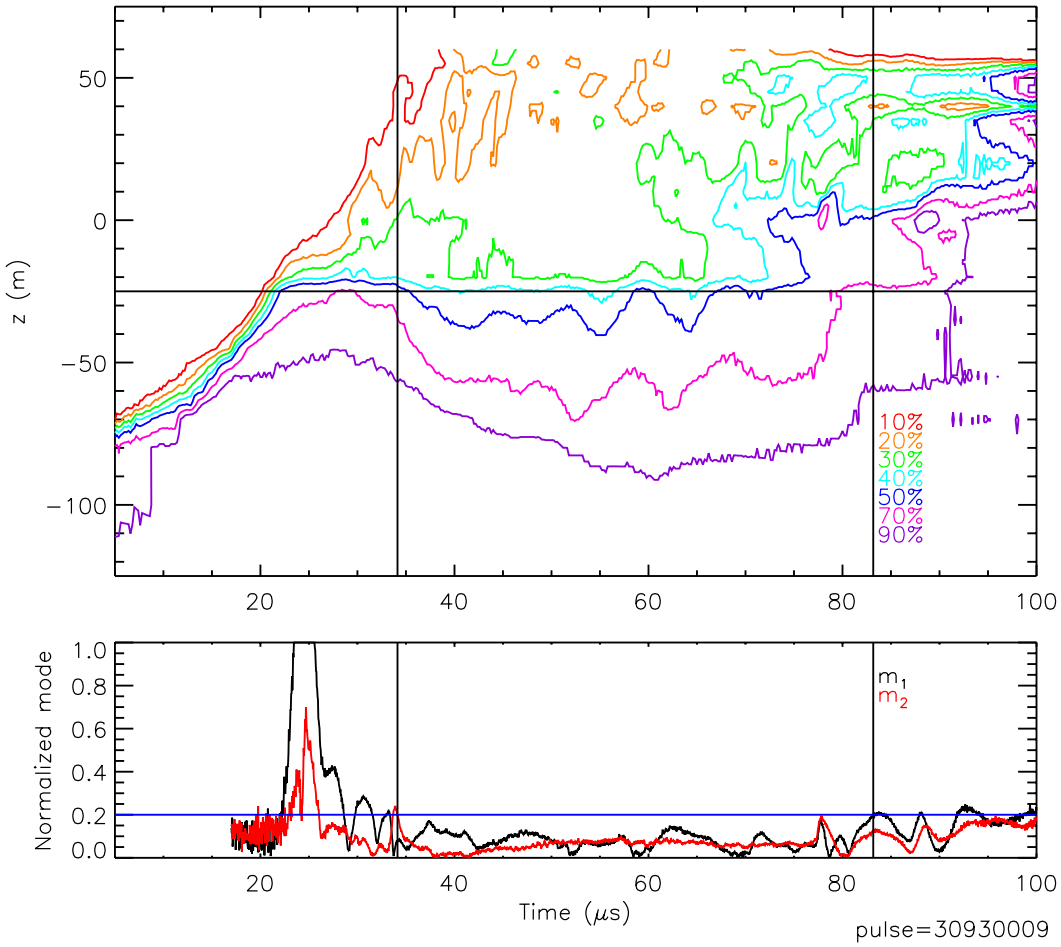

Figure 7.12: Contours of the magnetic field normalized by the current showing the axial dependence of the current. The horizontal line is at the end of the acceleration region. Below this line the contours represent axial current attachment locations. Above the line the contours are used to follow perturbations ion the Z-pinch. The horizontal lines indicate the beginning and end of the quiescent period. The mode activity increases as the current exits the acceleration region.

may be from an oscillatory instability in the ionization region, described by Morozov [42]. The axial motion of these lines indicate the motion of the deflagration ionization front. During the deflagration mod, plasma from the accelerator is being fed into the Z-pinch. As the connection points move towards the assembly region, the mode activity increases. The primary source of plasma is no longer driving the velocity shear.

This model has been used to increase the length of the quiescent period. The gas puff timings used in the nose cone studies were chosen to fill the acceleration region with neutral gas. A static fill cannot be used since the current sheet will form at the insulator. All the neutral gas would then be entrained in the current sheet, emptying the acceleration region and slowing the current sheet. The pressure near the gas injection plane is then increased by puffing four outer valves later in time. The breakdown of the gas occurs in the high pressure region. The plasma then evolves as was described. These studies have the longest quiescent periods. Filling the acceleration region behind the breakdown enables the deflagration to be maintained. This heuristic picture can be used to design a steady-state Z-pinch.

Chapter 8

CONCLUSIONS

The ZaP experiment is studying sheared flow stabilization of plasma instabilities. To test this theoretical prediction, a unique machine has been built. The experimental hardware has allowed for safe and steady operation, with a short turn around time between pulses. As the understanding of the experiment has increased, modifications have been made, which have increased the performance of the experiment.

A broad range of diagnostics are available on the experiment. The suite of diagnostics measures the plasma parameters necessary to verify that $\partial V_z/\partial r \geq 0.1 k V_A$. Like most projects, the ZaP experiment uses modern technology to improve existing diagnostic techniques. New analysis techniques improve information gained from the diagnostics. Two new methods are used to calculate the plasma properties. The magnetic field at the probes from perturbations of the current are found without calculating the induced surface currents. A new technique is used to deconvolve spectral intensities. The assumptions have been verified with experimental measurements. The impurities have the same parameters as the bulk plasma. The present diagnostics have been improved to reliably measure the plasma properties on every pulse.

One of the goals of the experiment is to study fusion grade plasmas. The initial conditions of the experiment affect the performance of the experiment. The plasmas made in the experiment are long lived and clean under certain operating conditions. The impurity and neutral content of the Z-pinch is low. Throughout the quiescent period heating is seen in the Z-pinch. The particle confinement time is consistent with coronal equilibrium. By using various experimental configurations and driving circuits the lifetime of the Z-pinch has been increased. With the present understanding and performance of the machine, the temperature and lifetime of the Z-pinch can be increased.

A heuristic understanding of the experiment has increased the performance of the ex-

periment. The behavior of the acceleration region agrees with previous experiments. The Z-pinch formation process is consistent with the model. A stable Z-pinch is on the axis of the machine throughout the quiescent period and a sheared flow is present during this time. The deflagration process in the accelerator maintains the flow in the Z-pinch. When the deflagration mode and flow decrease, instabilities are seen in the Z-pinch. Operational changes using this model have increased the lifetime of the Z-pinch.

The effect of flow shear stabilization has been correlated on the experiment. The evolution of the plasma in the experiment has two periods of high mode activity. During these periods, the velocity in the plasma is uniform. A sheared flow is measured during the quiescent period. Exact comparisons to the theory are not possible at this time, since the equilibrium pressure profile is unknown, stability thresholds for the velocity profile cannot be found. The behavior of the plasma is consistent with the theory described in Ref. [58].

Chapter 9

FUTURE WORK

The ZaP experiment has used a sheared flow to sustain an otherwise unstable configuration. The main study has not been completed. More work needs to be done to verify that the measured sheared flows stabilize the sausage and the kink mode in the Z-pinch. This requires new diagnostics, which can measure the pressure profile. While the present study continues, applications of the experiment beyond the basic physics study should be investigated. The applications can help guide the direction taken on the ZaP experiment. They can also lead to new devices which may ultimately become an economic source of electricity.

9.1 Improvements on the ZaP Experiment

Future work on the ZaP project will include measuring the pressure profile. The diagnostics presently used either measure chord integrated data and use deconvolution methods to calculate the plasma parameters. The pressure profile can be measured locally with Thomson scattering. Initial work has been performed on this system. Another method of measuring the pressure profile is with Zeeman splitting. More analysis of the data from this measurement is needed since it is a chord integrated measurement and the effect is small. Once the pressure profile is known, simulations of the Z-pinch can then determine the sheared flow threshold.

Another diagnostic which would aid with parameter studies is a multi chord IDS system. Presently the time evolution of the velocity profile is obtained by taking ICCD data at different times on the pulses throughout a day of operation. The evolution of the velocity profile is then found using pulses with similar mode characteristics. A multi-chord IDS system would significantly decrease the number of pulses required to obtain a time history of the radial velocity profile, which presently takes about 50-60 pulses. This system would

measure evolution of the velocity profile throughout the entire pulse. The velocity shear threshold could be obtained for multiple experimental configurations with this diagnostic.

9.2 Applications

Studying sheared flow stabilization of plasma is important for the fusion effort. The most important aspect of this research is the additional understanding of basic plasma physics that is gained. A sheared flow Z-pinch will simplify the design of a reactor, since there are no externally applied fields. The weight savings and inherent flow also make this configuration an attractive high thrust, high specific impulse, I_{sp}, thruster. These applications need scaling information to determine the operating parameters of the devices. Studies are needed on steady state Z-pinches, which approach fusion conditions.

9.2.1 Nuclear Power Plant

The main purpose for studying magnetic confinement of plasma is energy generation. A sheared flow Z-pinch is an attractive configuration for a fusion power plant. The flow through Z-pinch is a high beta machine. All of the magnetic energy is used to compress the plasma. There are no external magnetic fields needed for confinement nor stability. Conceptual designs for a reactor have been done by C.W. Hartman *et al.* and A.A. Newton *et al.* [45, 24]. A 10 keV plasma with a number density of 10^{26} m^{-3} would have to be at least 1 m long, the present length of the Z-pinch in ZaP. Fueling would come from plasma streaming in from the acceleration region and fusion would occur in the Z-pinch. The hole placed in the center of the end wall electrode allows the high energy ash to be naturally exhausted from the confinement region. Since the plasma is moving at a high velocity, a segmented MHD generator could be used at the exit to slow the plasma and generate electricity. This is one of the few configurations which could use the hot ash directly to generate electricity. Other configurations must first remove the ash through a vacuum pump or use the coolant from the walls to generate electricity. The hot gas from the sheared flow Z-pinch can also be used to run steam generators.

9.2.2 Space Thruster

ZaP could also be made into a high specific impulse and large thrust rocket. Z-pinches have many advantages over other magnetic confinement devices. Z-pinches are linear devices without external magnetic fields. The lack of externally applied magnetic fields reduces the mass of the system compared to other magnetic confinement devices. The greatest problem with this design has always been gross instabilities. ZaP is attempting to solve this problem by verifying the stabilizing effect of sheared flows.

The extension of the ZaP experiment to a thruster has been studied by U. Shumlak [56]. The endwall would be replaced with a nozzle. The bulk of the plasma in the Z-pinch moves away from the inner electrode with a high velocity. Aneutronic fuels could be completely burned in a 1.5 m to 18 m long Z-pinch. The portion of the plasma which enters the inner electrode would be used in an energy converter to supply power to the thruster. The large thrust and high specific impulse from such a thruster, may make manned deep space exploration possible.

9.2.3 Steady State Operation

The ZaP experiment has been funded as a basic physics experiment. It has applications to fusion energy and space propulsion. This could require steady state operation. The present model of the phases of the experiment show that steady state operation of this configuration is possible. Shown in Fig. 9.1 is a diagram of how the experiment can run in steady state. Neutral gas is constantly flowing into the acceleration region. A deflagration mode will supply plasma to the Z-pinch. The stability is provided by the sheared flow. The pressure is maintained by supplying a constant current to the experiment. The plasma is then exhausted through the end wall. This mode of operation is only limited by the gas injection rate, pumping speed, and driving circuit, not instabilities.

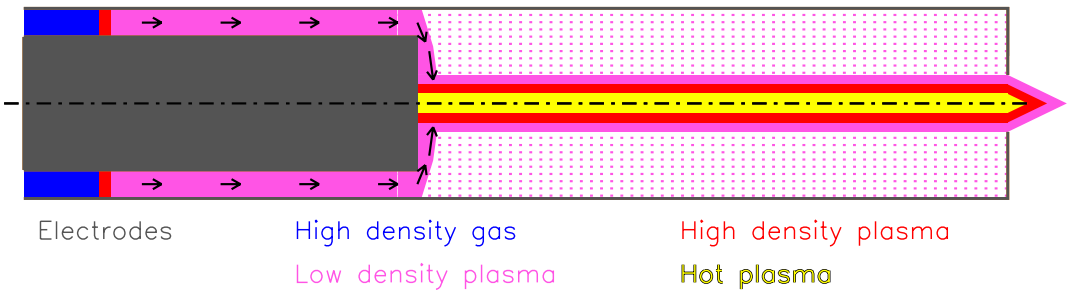

Figure 9.1: A sheared flow Z-pinch can be run in steady state. The deflagration mode is sustained by constantly injecting neutral gas into the acceleration region. Current in the Z-pinch would provide the containment. The Z-pinch is stabilized with the generated sheared flow.

BIBLIOGRAPHY

[1] T. D. Arber and D. F. Howell. The effect of sheared axial flow on the linear stability of the z-pinch. *Physics of Plasmas*, **3**(2):554–560, February 1995.

[2] R. E. Bell. An inversion technique to obtain full poloidal velocity profiles in a tokamak plasma. *Review of Scientific Instruments*, **68**(2):1273–1280, February 1997.

[3] P. R. Bevington and D. K. Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. WCB McGraw-Hill, Boston, Massachusetts, second edition, 1992.

[4] A. Brablec, D. Trunec, and F. Stastny. Deconvolution of spectral line profiles: Solution of the inversion problem. *Journal of Physics D: Applied Physics*, **32**:1870–1875, 1999.

[5] J. E. Bright. Magnetic mode analysis in the ZaP flow-stabilized Z-pinch experiment. Master's thesis, University of Washington, 2002.

[6] K. V. Brushlinskii, A. M. Zaborov A. N. Kozlov, A. I. Morozov, and V. V. Savel'ev. Numerical modeling of plasma flow in high-current quasistationary plasma accelerators. *Fizika Plazmy*, **16**(2):147–57, February 1990.

[7] K. V. Brushlinskii and A. I. Morozov. Calculation of two-dimensional plasma flows in channels. *Reviews of Plasma Physics*, **8**:105, 1974.

[8] C. J. Buchenauer and A. R. Jacobson. Quadrature interferometer for plasma density measurements. *Review of Scientific Instruments*, **48**(7):769–774, July 1977.

[9] T. D. Butler and J. L. Cook. Numerical analysis of a coaxial accelerator. *Physics of Fluids*, **11**:2286–2288, 1968.

[10] T. D. Butler, I. Henins, F. C. Jahoda, J. Marshall, and R. L. Morse. Coaxial snowplow discharge. *Physics of Fluids*, **12**(9), 1904-1916 1969.

[11] T. D. Butler, I. Henins, F. C. Jahoda, J. Marshall, R. L. Morse, J. Marshall, and R. L. Morse. Coaxial snowplow discharge, page 35-48. Status Report LA-4075-MS, Los Alamos Scientific Laboratory of the University of California, October 1968.

[12] P. G. Carolan and V. A. Piotrowicz. The behavior of impurities out of coronal equilibrium. *Plasma Physics*, **25**(10):1065–1086, 1983.

[13] F. Castillo, M. Milanese, R. Moroso, and J. Pouzo. Evidence of thermal and non-thermal mechanisms coexisting in dense plasma focus D-D nuclear reactions. *Journal of Physics D: Applied Physics*, **33**(2):141–147, January 1999.

[14] D. Y. Cheng. Plasma deflagration and the properties of a coaxial plasma deflagration gun. *Nuclear Fusion*, **10**:305–317, 1970.

[15] G. D. Conway, D. N. Borba, B. Alper, D. V. Bartlett, C. Gormezano, M. G. von Hellermann, A. C. Maas, V. V. Parail, P. Smeulders, and K-D. Zastrow. Suppression of plasma turbulence during optimized shear configurations in jet. *Physical Review Letters*, **84**(7):1463–1466, February 2000.

[16] S. DeSouza-Machado, A. B. Hassam, and R. Sina. Stabilization of z pinch by velocity shear. *Physics of Plasmas*, **7**(11):4632–4643, November 2000.

[17] A. Engel, K. N. Koshelev, Yu. V. Sidelnikov, S. S. Churilov, C. Gavrilescu, , and R. Lebert1. Population inversion due to charge-exchange interaction of plasma jets from plasma focus with residual gas. *Physical Review E*, **58**(6):7819 7822, December 1998.

[18] A. B. Filuk and J. E. Bailey. Analyzing noisy spectral line shapes. *Review of Scientific Instruments*, **63**(10), 4783-4785 1992.

[19] J. P. Friedberg. *Ideal Magnetohydrodynamics*. Plenum Press, 233 Spring Street, New York, New York 10013 USA, first edition, 1987.

[20] R. P. Golingo and U. Shumlak. Spatial deconvolution technique to obtain velocity profiles from chord integrated spectra. *Review of Scientific Instruments*, **74**(4):2332–2337, April 2003.

[21] H. R. Griem. *Plasma Spectroscopy*. McGraw-Hill Book Company, New York, New York, 1964.

[22] M. G. Haines and M. Coppins. Universal diagram for regimes of z-pinch stability. *Physical Review Letters*, **66**(11):1462–1465, March 1991.

[23] M. G. Haines, S. V. Lebedev, J. P. Chittenden, F. N. Beg, S. N. Bland, and A. E. Dangor. The past, present, and future of z pinches. *Physics of Plasmas*, **7**(5):1672–1680, May 2000.

[24] C. W. Hartman, G. Carlson, M. Hoffman, R. Werner, and D.Y. Cheng. A conceptual fusion reactor based on the high-plasma-density z-pinch. *Nuclear Fusion*, **17**(5):909–18, October 1977.

[25] D. J. Den Hartog and R. P. Golingo. Telecentric viewing system for light collection from a z-pinch plasma. *Review of Scientific Instruments*, **72**(4):2224–2225, April 2001.

[26] I. Henis, J. Marshall, and A. A. Newton. The continuous flow pinch investigation page 23-28. Status Report LA-4075-MS, Los Alamos Scientific Laboratory of the University of California, Oct 1968.

[27] I. H. Hutchinson. *Principles of Plasma Diagnostics*. Cambridge University Press, New York, NY, 1987.

[28] I. H. Hutchinson, R. S. Granetz, A. Hubbard, J. A. Snipes, T. Sunn Pedersen, M. Greenwald, B. LaBombard, and the Alcator Group. Edge transport barrier phenomena in Alcator C-MOD. *Plasma Physics and Controlled Fusion*, **41**:A609–A616, March 1998.

[29] A. R. Jacobson. Heterodyne quadrature interferometer for simultaneous measurements of plasma density along several chords. *Review of Scientific Instruments*, **49**(5):673–674, May 1978.

[30] G. V. Karpov, E. N. Smirnov, and V. N. Suvorov. Axial motion of current shell in a dense plasma focus. *Soviet Physics- Technical Physics*, **21**(3):293–296, 1976.

[31] W. Kies, G. Decker, U. Berntien, Yu. V. Sidelnikov, D. A. Glushkov, K. N. Koshelev, D. M. Simanovskii, and S. V. Bobashev. Pinch modes produced in the speed2 plasma focus. *Plasma Sources Science and Technology*, **9**(3):279–287, August 2000.

[32] R. Kristal and R. W. Paterson. Bragg cell heterodyne interferometry of fast plasma events. *Review of Scientific Instruments*, **47**(11):1357–1359, November 1976.

[33] M. Kruskal and M. Schwarzschild. Some instabilities of a completely ionized plasma. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, volume 223 of *A*, pages 348–360. The Royal Society, 1954.

[34] J.D. Lawson. Some criteria for a power producing thermonuclear reactor. In *Proceedings of the Physical Society (London)*, volume B70, page 46, 1957.

[35] S. Lee, T. Y. Tou, S. P. Moo, M. A. Eissa, A. V. Gholap, K. H. Kwek, S. Mulyodrono, A. J. Smith, Suryadi, W. Usada, and M. Zakaullah. A simple facility for the teaching of plasma dynamics and plasma nuclear fusion. *American Journal of Physics*, **56**(1):62–68, January 1988.

[36] M. F. Lu. Instability of the current sheath implosion in an 18-kJ Mather-type plasma focus. *Physical Review E*, **54**(2):R1074 R1077, August 1996.

[37] J. Marshall. Performance of a hydromagnetic plasma gun. *Physics of Fluids*, **3**(1):134–135, January-February 1960.

[38] 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: http://physics.nist.gov/asd.

[39] J. W. Mather. Formation of a high-density deuterium plasma focus. *Physics of Fluids*, **8**(2):366–377, February 1965.

[40] M. Mathuthu and T. G. Zengeni. Measurement of magnetic field and velocity profiles in 3.6 kJ United Nations University/International Center for Theoretical Physics Plasma Focus Fusion Device. *Physics of Plasmas*, **3**(12):4572–4576, December 1996.

[41] K.J. McCollam. *Investigation of Magnetic Relaxation in Coaxial Helicity Injection.* PhD thesis, University of Washington, 2000.

[42] A. I. Morozov. Principles of coaxial (quasi-) steady-state plasma accelerators. *Fizika Plazmy*, **16**(2):131–46, February 1990.

[43] A. I. Morozov and L. S. Solov'ev. Steady-state plasma flow in a magnetic field. *Reviews of Plasma Physics*, **8**:1, 1974.

[44] A. Muravich. Dynamics of dense sheet z-pinch. *Journal of the Physical Society of Japan*, **69**(6):1692–1704, Jun. 2000.

[45] A. A. Newton. The continuous flow pinch as a thermonuclear reactor, page 31-35. Status Report LA-4075-MS, Los Alamos Scientific Laboratory of the University of California, October 1968.

[46] C. Palmer. *Diffraction Grating Handbook.* Spectra-Physics, Inc., 705 St. Paul Street, Rochester, New York 14605 USA, 2003.

[47] R. E. Peterkin and M. H. Frese. *MACH: A Reference Manual*. Air Force Research Laboratory, Kirtland AFB, New Mexico, first edition, 1998.

[48] G. M. Petrov. A simple algorithm for spectral line deconvolution. *Journal of Quantitative Spectroscopy and Radiative Transfer*, pages 281–287, 2002.

[49] W. F. Ray and C. R. Hewson. High performance Rogowski current transducers. In *Conference Record of the 2000 IEEE*, volume 5, pages 3083–3090. IEEE, October 2000.

[50] W. F. Ray and C. R. Hewson. Practical aspects of Rogowski current transducer performance. In *PCIM Conference 2001, Nurenburg*, volume ISBN 3-928643-27-4. PCIM, 2001.

[51] E. L. Ruden. Rayleigh-Taylor instability with a sheared flow boundary layer. *IEEE Transactions on Plasma Science*, **30**(2):611–615, April 2002.

[52] J. Scheffel and M. Coppins. Reconsideration of the m = 0 z-pinch stability. *Nuclear Fusion*, **33**(1):101–115, 1993.

[53] E. Sevillano. *Magnetic reconnection in field-reversed configurations*. PhD thesis, University of Washington, 1983.

[54] V. D. Shafranov. *Atomic Energy*, **5**:38, 1956.

[55] T. Shreve. Characterization of a pulse valve. Technical Report 20030527:62-7316, University of Washington, Aerospace & Energetics Research Program, University of Washington, Seattle, WA 98195, May 2003.

[56] U. Shumlak. A near-term, z-pinch fusion space thruster. *AIAA*, (2000-3368), 1999.

[57] U. Shumlak, R. P. Golingo, B. A. Nelson, and D. J. Den Hartog. Evidence of stabilization in the Z-pinch. *Physical Review Letters*, **87**(20):205005, 2001.

[58] U. Shumlak and C.W. Hartman. Sheared flow stabilization of the m=1 kink mode in z pinches. *Physical Review Letters*, **75**(18), 3285-8 1995.

[59] U. Shumlak, B. A. Nelson, R. P. Golingo, S. L. Jackson, E. A. Crawford, and D. J. Den Hartog. Sheared flow stabilization experiments in the zap flow Z pinch. *Physics of Plasmas*, **10**(5):1683–1690, May 2003.

[60] V. I. Sotniklov, I. Paraschiv, V. Makhin, B. S. Bauer, J. N. Leboeuf, and J. M. Dawson. Linear analysis of sheared flow stabilization of global magnetohydrodynamic instabilities on the Hall fluid model. *Physics of Plasmas*, **9**(3):913–922, March 2002.

[61] J. A. Stillerman, T. W. Fredian, K. A. Klare, and G. Manduchi. MDSplus data acquisition system. *Review of Scientific Instruments*, **68**(1):939–942, January 1997.

[62] E. J. Synakowski. Formation and structure of internal and edge transport barriers. *Plasma Phys. Control. Fusion*, **40**(5):581–596, May 1998.

[63] H. Sze, P. L. Coleman, B. H. Failor, A. Fisher, J. S. Levine, Y. Song, E. M. Waisman, J. P. Apruzese, Y. K. Chong, J. Davis, F. L. Cochran, J. W. Thornhill, A. L. Velikovich, B. V. Weber, C. Deeney, C. A. Coverdale, and R. Schneider. Initial results for an argon z pinch using a double-shell gas puff. *Physics of Plasmas*, **7**(10):4223–4226, October 2000.

[64] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia,PA, 1997.

[65] A. Y. Voloshko, I. E. Garkusha, O. E. Kazakov, A. I. Morozov, O. S. Pavlichenko, D. G. Solyakov, V. I. Tereshin, M. A. Tiarov, S. A. Trubchaninov, A. V. Tsarenko, and V. V Chebotarev. Investigation of a two-step quasistationary plasma accelerator with rod-type electrodes. *Fizika Plazmy*, **16**(2):158–67, February 1990.

[66] A. Y. Voloshko, I. E. Garkusha, A. I. Morozov, D. G. Solyakov, V. I. Tereshin, A. V. Tsarenko, and V. V. Chebotarev. Investigation of the local plasma flow pattern in a two-step P-50 quasistationary plasma accelerator. *Fizika Plazmy*, **16**(2):168–75, February 1990.

[67] M. G. von Hellermann and H. P. Summers. Atomic modeling and spectroscopic diagnostics (invited). *Review of Scientific Instruments*, **63**(10):5132–5139, December 1992.

[68] F. Wagner, G. Becker, K. Behringer, D. Campbell, A. Eberhagen, W. Engelhardt, G. Fussmann, O. Gehre, J. Gernhardt, G. v. Gierke, G. Haas, M. Huang, F. Karger, M. Keilhacker, O. Kluber, M. Kornherr, K. Lackner, G. Lisitano, G. G. Lister, H. M. Mayer, D. Meisel, E. R. Muller, H. Murmann, H. Niedermeyer, W. Poschenrieder, H. Rapp, H. Rohr, F. Schneider, G. Siller, E. Speth, A. Stabler, K. H. Steuer, G. Venus, O. Vollmer, and Z. Yu. Regime of improved confinement and high beta in neutral-beam-heated divertor discharges of the ASDEX tokamak. *Physical Review Letters*, **49**(19):1408 1412, November 1982.

[69] D. M. Woodall and L. K. Len. Observation of current sheath transition from snow plow to deflagration. *Journal of Applied Physics*, **57**(3), 961-964 1985.

[70] M. Zakaullah, Imtiaz Ahmad, A. Omar, G. Murtaza, and M. M. Beg. Effects of anode shape on plasma focus operation with argon. *Plasma Sources Sci. Technology.*, **5**(3):544–552, August 1996.

Appendix A

MAGNETIC FIELD IN A FLUX CONSERVER

A.1 Introduction

Often in magnetically confined plasma experiments, magnetic probes measure the magnetic field at the surface of a flux conserver. The magnetic field for an ideally located plasma current is well understood. When the plasma current is displaced, the magnetic field tries to penetrate the conducting walls. Surface currents are created in the walls, which keep the flux constant. The fields measured at the wall are a sum of the fields from the plasma current and the surface currents. While the magnetic field at the wall from the plasma current is easily calculated, the calculation of the magnetic field from the surface currents in the wall is difficult due the coupling of the currents and the magnetic field. A simple relationship between the surface magnetic field and the plasma current is derived. This relationship does not require the calculation of the surface currents, saving time when analyzing surface magnetic measurements.

On the ZaP experiment, a Z-pinch is formed inside a copper tube. The magnetic field is measured at various locations along the wall. When the plasma current is located on the z axis, the surface magnetic field, B_θ, is given by $B_\theta = \mu_0 I_p / 2\pi r_{wall}$ where I_p is the plasma current and r_{wall} is the wall radius. The Z-pinch moves on time scales faster than the soak through time of the copper. As the plasma moves, surface currents are created along the wall which keep the normal component of the magnetic field zero. The total magnetic field for the displaced current is measured by the magnetic probes in the wall. Relating the measured magnetic fields to the displacement of the current is time consuming when the distribution of the surface currents is calculated. It can be shown that the surface magnetic field can be calculated by using the magnetic field of unperturbed current and the magnetic field of a loop of current which models the perturbation. With this method, the distribution

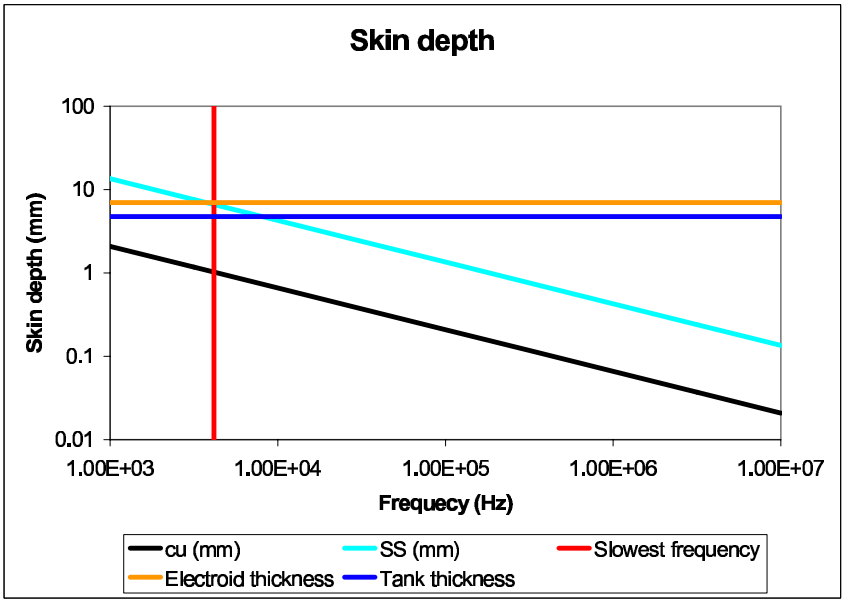

Figure A.1: Shown are the skin depth of copper (black) and stainless steel (cyan) for the frequency range of the ZaP experiment. The horizontal lines are the thicknesses of the tank (blue) and the outer electrode (orange). The vertical line corresponds to the slowest frequency of the experiment. Since the skin depth is smaller than the thickness of the outer electrode, the electrode acts as a flux conserver.

of the surface currents do not need to be found to understand the measured magnetic fields.

This method is described in the next section. The A.3 section shows the agreement of the method when the surface current is found for different types of displacements. The A.4 section shows comparisons with experiments. The final section is the conclusion.

A.2 Theory

As kinks form along the Z-pinch the current is displaced from the z axis. This changes the magnetic field along the outer electrode. The magnetic field changes on time scales faster than the soak through time of the copper electrode, show in Fig. A.1. The large frequencies justify the perfect flux conserver assumption used in this analysis.

The perturbation of the current is modeled as an axial current and as a loop of current, which follows the perturbation and returns along the z axis in the opposite direction as the

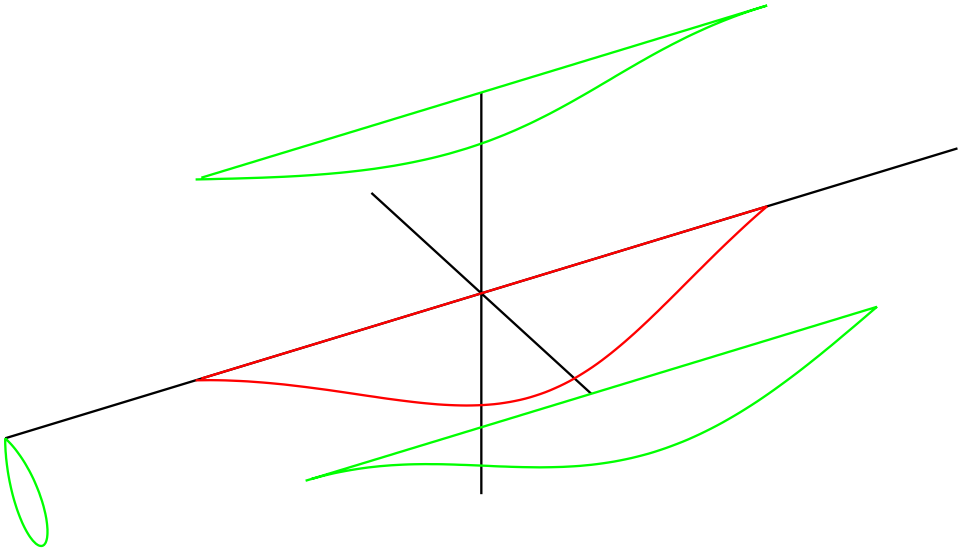

Figure A.2: A loop of current (red) used to model the perturbation to the Z-pinch. The axis are shown as black lines. Projections of the current onto the planes are shown as green lines.

axial current, shown in Fig. A.2. The magnetic field at the inner surface of the wall, $\mathbf{B}|_{\text{wall}}$, is given by

$$\mathbf{B}|_{\text{wall}} = \mathbf{B}^{(0)}|_{\text{wall}} + \mathbf{B}^{(1)}|_{\text{wall}} + \mathbf{B}^{(2)}|_{\text{wall}} \tag{A.1}$$

where $\mathbf{B}^{(0)}$ is the magnetic field from the axial current, $\mathbf{B}^{(1)}$ is the magnetic field from the loop current and $\mathbf{B}^{(2)}$ is the magnetic from surface currents, $j^{(2)}$, on the wall. The wall subscript has been dropped since all of the fields are measured at the wall. The boundary conditions, B.C., at the wall for a perfect conductor are

$$\mathbf{B}_{\perp}^{(1)} + \mathbf{B}_{\perp}^{(2)} = 0 \qquad \text{B.C. 1: at the wall of the flux conserver} \tag{A.2}$$

$$\mathbf{B}^{(1)} + \mathbf{B}^{(2)} = 0 \qquad \text{B.C. 2 : outside the flux conserver} \tag{A.3}$$

The magnetic field at any location along the electrode can be found by drawing an Amperian loop around an element of the induced wall current. The surface current is centered in the element, shown in Fig. A.3. Using the B.C.s and letting the thickness of the element goto

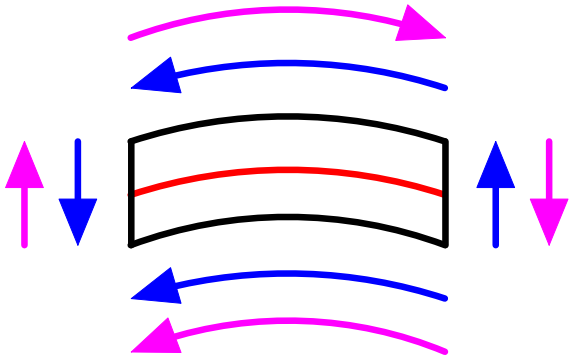

Figure A.3: Shown is an edge view of an element of the outer electrode with the magnetic fields. The element is black, $j^{(2)}$ is red, $\mathbf{B}^{(1)}$ is blue and $\mathbf{B}^{(2)}$ is magenta. The arrows show the direction of the fields. The current is placed in the center of the element. Th magnetic fields are calculated at the corners of the element.

zero, $\mathbf{B}^{(2)}$ can be written in terms of $\mathbf{B}^{(1)}$

$$\mathbf{B}_{\perp}^{(2)} = -\mathbf{B}_{\perp}^{(1)} \tag{A.4}$$

$$\mathbf{B}_{\parallel}^{(2)} = \mathbf{B}_{\parallel}^{(1)} \tag{A.5}$$

The total magnetic field inside the wall is now found without calculating the surface currents

$$\mathbf{B}_{\parallel} = \mathbf{B}_{\parallel}^{(0)} + 2\mathbf{B}_{\parallel}^{(1)} \tag{A.6}$$

$$\mathbf{B}_{\perp} = \mathbf{B}_{\perp}^{(0)} \tag{A.7}$$

. $\mathbf{B}_{\parallel}^{(1)}$ is found with the Biot-Savart law.

A.3 Calculation of the Surface Currents

Equation A.7 is verified by calculating the surface magnetic fields, including the surface currents from simulated displacements. The outer electrode is divided into elements shown in Fig. A.3. The current is placed in the center of the element and the magnetic field is calculated at the corners. The thickness of the element is kept larger than the maximum spacing of the currents to remove localized concentrations of the magnetic field caused when the current is discretized. The displaced currents are defined such that the original magnetic

and the perturbed magnetic field decouple. Ampere's law can be written as

$$\nabla \times \left(\mathbf{B}^{(0)} + \mathbf{B}^{(1)} + \mathbf{B}^{(2)}\right) = \mu_0 \left(\mathbf{j}^{(\mathbf{0})} + \mathbf{j}^{(\mathbf{2})}\right) \tag{A.8}$$

where $\mathbf{j}^{(\mathbf{0})}$ is the original surface current of the wall and $\mathbf{j}^{(\mathbf{2})}$ is the additional surface current of the wall needed to keep the flux constant. Since the $\mathbf{B}^{(0)}$ and $j^{(0)}$ terms cancel, the $\mathbf{B}^{(2)}$ terms are unknown and derivatives with respect to r are infinite for a thin surface current, the surface currents are found iteratively using Eq. A.8. The (0), (2), and d/dr terms are neglected giving

$$\triangle \mathbf{j}^{(\mathbf{2})} = \frac{1}{\mu_0} \nabla \times \mathbf{B}_r^{(1)} \tag{A.9}$$

where $\triangle \mathbf{j}^{(\mathbf{2})}$ is the additional surface current added at each iteration. Since B.C. 2 is not used, it serves as a check that the calculated surface currents are correct. Two types of displacements are studied, a constant displacement of the current away from the axis and sinusoidal perturbations along the axis.

A.3.1 Constant Displacements

A straight current, located of the off the axis of the electrode, will change the magnitudes of the return surface current but not the direction. The electrode is divided into infinitely long strips of width $ds = r_{\text{wall}}\ d\theta$. The resulting surface magnetic field is then given by

$$\boldsymbol{B}^{(\boldsymbol{j})} = \sum_i \frac{\mu_0 I^{(i)} \left(-y^{(ij)}\hat{i} + x^{(ij)}\hat{j}\right)}{2\pi \left(r^{ij}\right)^2} \tag{A.10}$$

where $B^{(j)}$ is the magnetic field at node j, $I^{(i)}$ is the current at location i, and $x^{(ij)}$, $y^{(ij)}$, $r^{(ij)}$ are the x, y, and total distances from the current to the node respectively. The surface current for the next iteration, $I^{(i)}_{(n+1)}$, is given by

$$I^{(i)}_{(n+1)} = I^{(i)}_{(n)} + \frac{2ds}{\mu_0 r_{\text{wall}}} \left(\frac{dB_r}{d\theta}\right)^{(i)}_{(n)} \tag{A.11}$$

The magnetic field is calculated with an IDL code, *find_b_wall.pro* Sec. C.2.5. The calculated magnetic field is shown in Fig. A.4. The theoretical magnetic field, dashed blue line, agrees with calculated magnetic field, black line. The magnetic field normal to and outside of the electrode is zero.

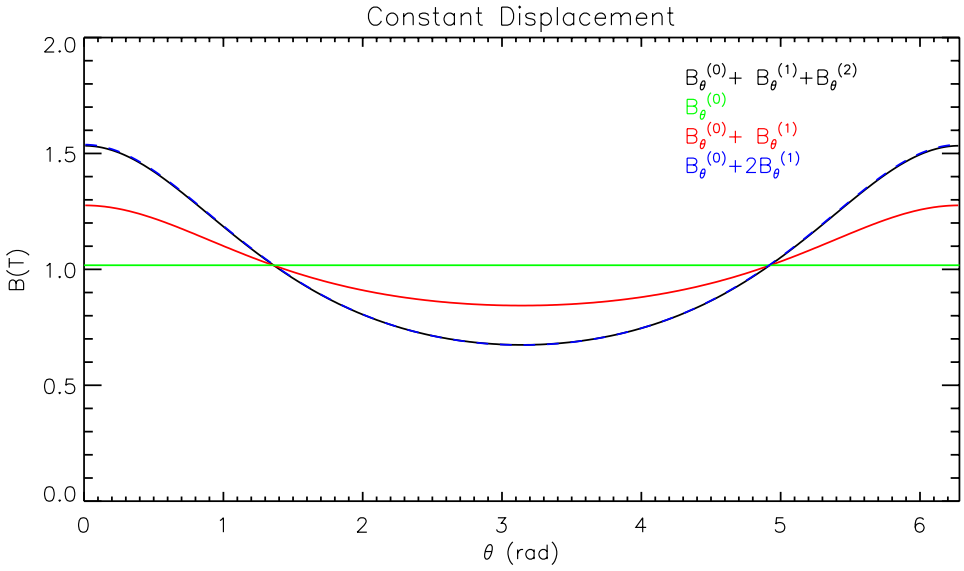

Figure A.4: The magnetic fields are calculated for a constant displacement of the current off axis. The different magnetic fields are defined in Eq. A.1. The calculated magnetic field from the code and the estimated magnetic field from Eq. A.6 are identical.

A.3.2 Sinusoidal Displacements

When the displacement of the current has a z dependence, axial magnetic fields and azimuthal surface currents are generated. The current is assumed to have the displacement, shown in Fig. A.2, given by

$$x(z) = A\cos\left(n_{\text{twist}}\pi\frac{z-l_s}{l_0}\right)\cos\left(n_{\text{osc}}\pi\frac{z-l_s}{l_0}\right) \tag{A.12}$$

$$y(z) = A\sin\left(n_{\text{twist}}\pi\frac{z-l_s}{l_0}\right)\cos\left(n_{\text{osc}}\pi\frac{z-l_s}{l_0}\right) \tag{A.13}$$

where A is the maximum radial displacement, l_s is the starting location of the perturbations, l_0 is the total length of the perturbations, n_{twist} is the number of azimuthal cycles, and n_{osc} is the number of oscillations. The current loop is completed with the current returning along the axis. The parameters which describe the perturbation in Fig. A.2 are $A = 0.05$, $n_{\text{twist}} = .5$, $n_{\text{osc}} = 1$, $l_s = 0.15$, and $l_0 = 0.30$. $B^{(1)}$ is used as the initial guess of the magnetic field. The surface current is found with Eq. A.9. The radial magnetic field from the perturbation and the surface currents needed to cancel the radial field are shown in

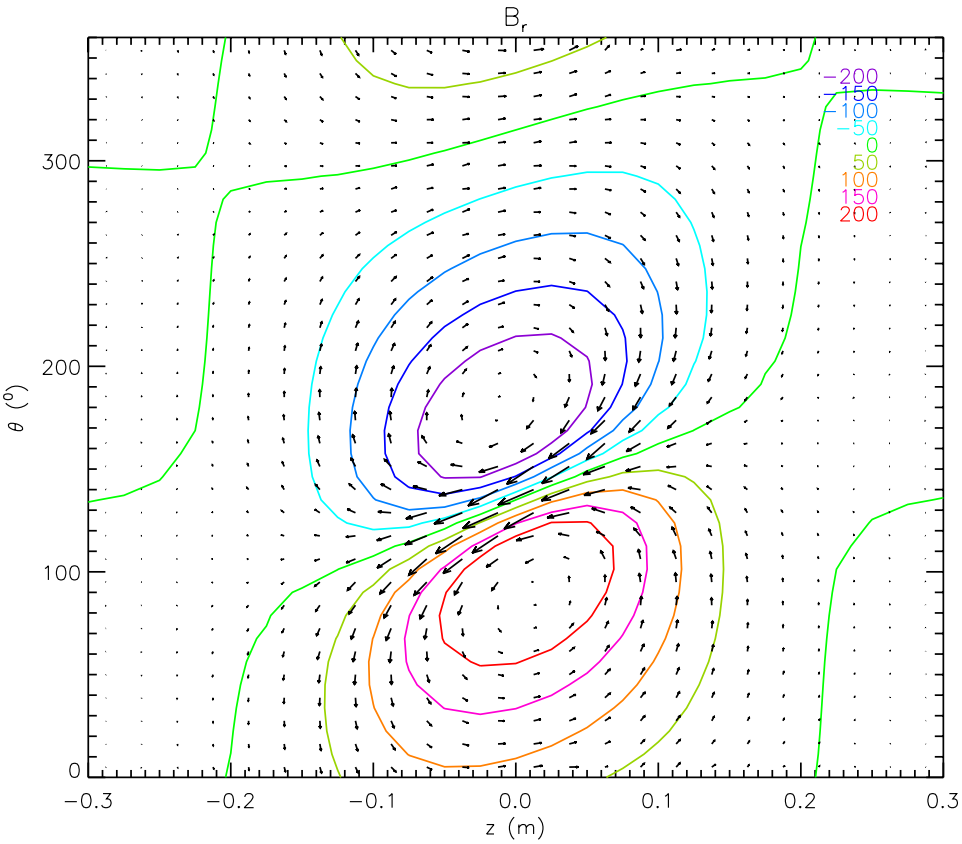

Figure A.5: The radial magnetic fields at the outer electrode are found using the Biot-Savart law. The surface currents which cancel $B_\perp$ are the arrows.

Fig. A.5. The total magnetic field at the wall is given by Eq. A.8. This process is repeated until $B_r = 0$. The total magnetic field from finding the surface currents and using Eq. A.6 are shown in Fig. A.6. The differences in the two fields are from B_r not going completely to zero and the size of the elements.

A.4 Experimental Verification

The magnetic field of an off-axis current is measured using the calibration rod. After the probes are calibrated the rod is moved off the axis. The procedure used to calibrate the probes is repeated. The magnetic field of a known displacement is then measured. The

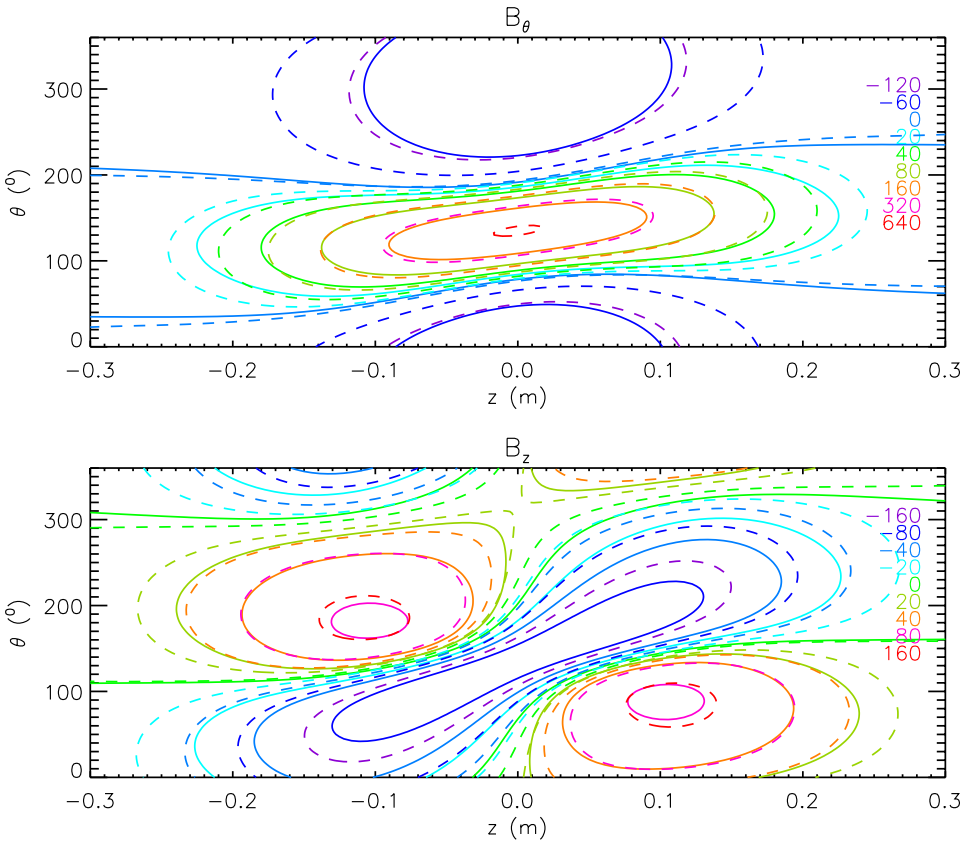

Figure A.6: Shown are the magnetic fields calculated for the displacement of the current shown in Fig. A.2 at the wall. The solid contours are magnetic fields calculated using the surface currents. The total magnetic field from the loop and surface currents are shown as dashed lines. The total tangential field is approximately twice the tangential field from the loop of current alone.

results are shown in Fig. A.7. The calculated magnetic fields from Eq. A.7 agree with the measured fields as the calibration rod is displaced.

A.5 Conclusion

This magnetic field can be found at the surface of a flux conserver without calculating the surface currents. This result has been verified computationally and experimentally. Not having to calculate the surface magnetic fields leads to a great saving of computational time.

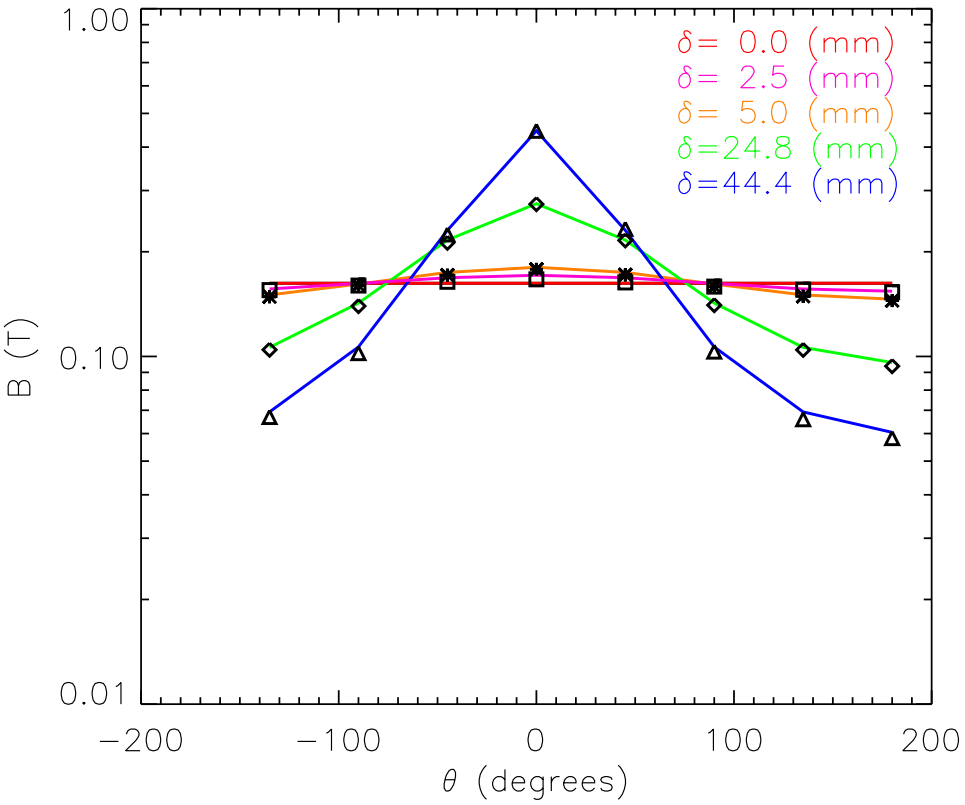

Figure A.7: Shown are the magnetic fields for an off-axis current. The colors correspond to the theoretic values. The symbols are the average magnetic field at each probe for four pulses. The displacement of the rod is shown in the upper right corner. The error bars are smaller than the symbols.

The method enables the quick calculation of the magnetic field measured at the probes for any perturbation of the current.

Appendix B

GAS PUFF STUDIES

The reproducible Z-pinches with a long quiescent period are only formed with the proper initial conditions. A survey of gas puff timings was performed from March 12, 2003, to May 5, 2003, to characterize the behavior of the machine. The outer electrode extension was just added and the current was driven with the pulse forming network. The number of 9 kV pulses taken at each gas puff timing are shown in Fig. B.1. At least four pulses were taken at each setting. Large changes in the settings were avoided throughout a day of running to keep wall conditions similar. Additional pulses were taken at many of the settings to improve the statistics of regions where surveys may be conducted.

The snowplow model predicts that the final velocity of the current will decrease when the neutral density in the accelerator is increased. By puffing the valves earlier in time, the arrival time of the current sheet at the $z = 0$ cm azimuthal array should be later. The calculation of velocity of the current sheet at the exit of the accelerator is affected by filaments. Since 441 pulses are used for this study, the arrival time at the $z = 0$ cm array, shown in Fig. B.2, is used to measure the velocity. As expected, the velocity of the current sheet is decreases. The outer gas puff valves timing has a larger effect on the velocity. The slopes of the 18 and 19 μs contours are less than 45 degrees. As the gas puff timing is moved earlier in time, gas escapes from the accelerator. The velocity of the current sheet has a maximum at the -1.7 ms inner gas puff and -1.0 ms outer gas puff timing. As the gas puff timings are moved earlier, the current sheet velocity increases.

This survey was conducted to evaluate the performance of the machine as the gas puff timings were changed. Two of the many ways used to characterize the Z-pinch are shown in Fig. B.3. The repeatability of the pulses has been related to the plasma current for this experimental configuration. The top contour shows that at the extreme of the gas puff timings the plasma current is not reproducible. Small differences of the plasma current are

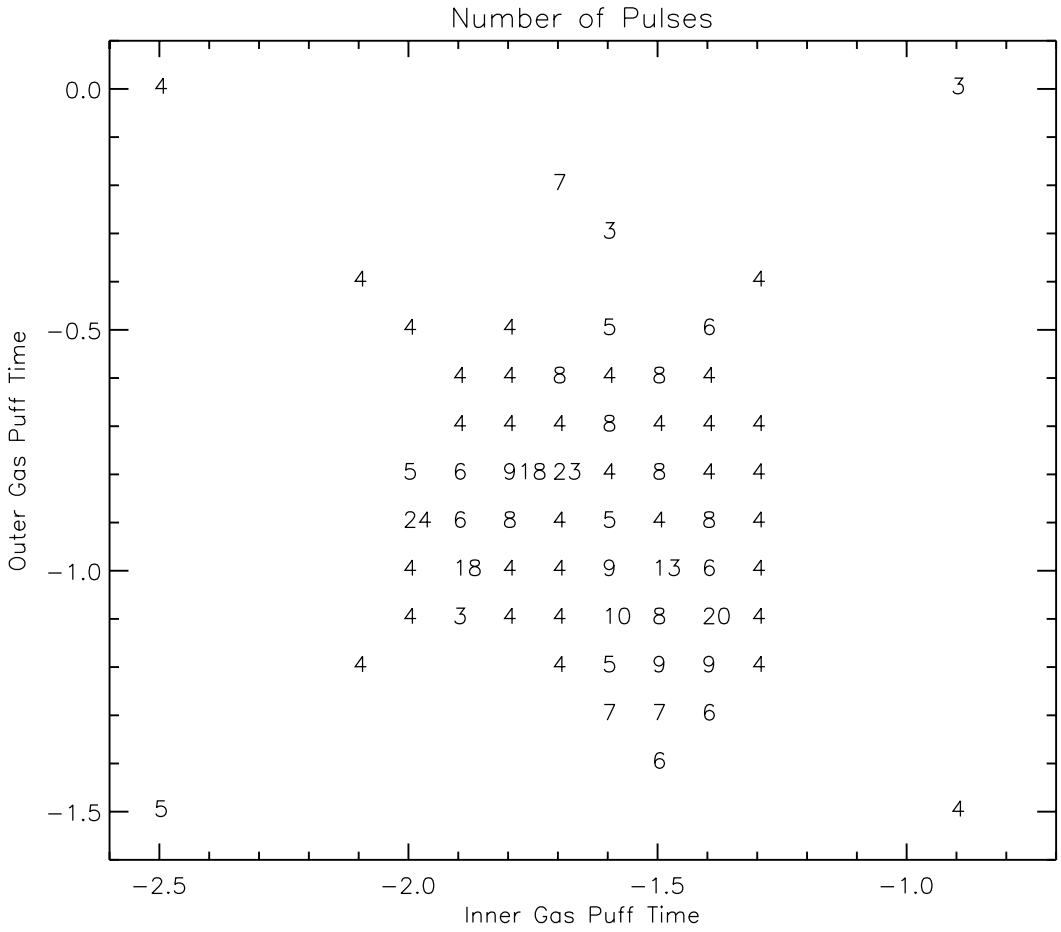

Figure B.1: The number of 9 kV pulses taken at each set of gas puff timing are shown. The eight outer valves were all puffed at the same time. The gas puff settings in the upper right corner should be avoided. The plasma is starved of current carriers for this gas puff setting.

measured for the normal operation region. Another method of characterizing a pulse is to look at the current exiting the hole in the outer electrode endwall. Increases in this current have been correlated to longer quiescent periods. Current exiting the endwall is also needed when characterizing end losses. The bottom contour shows optimal timings, which can be used for end loss studies.

The length of the quiescent period is one of the main figures of merit of the experiment. The average length of the quiescent period from two of the azimuthal arrays are shown

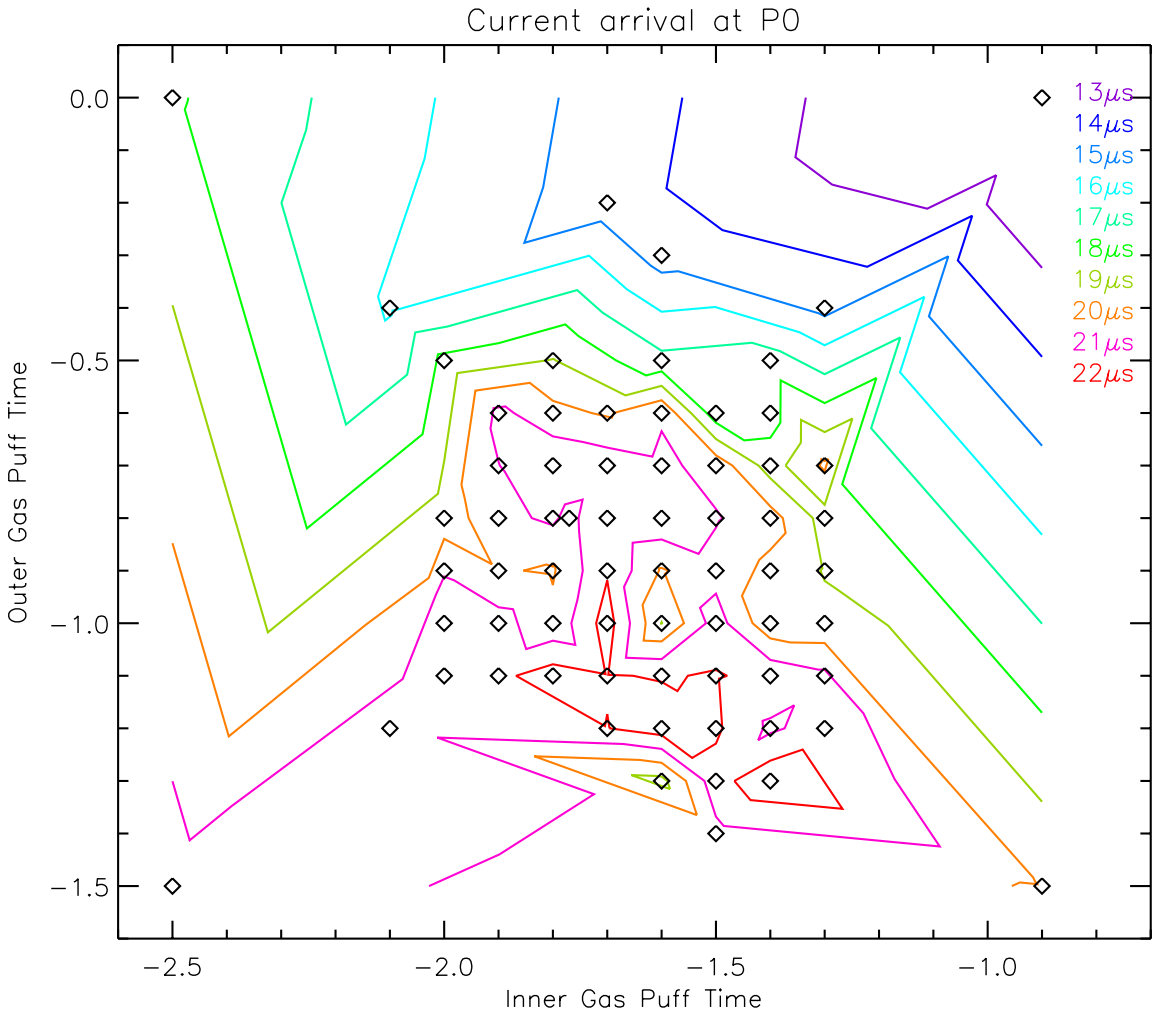

Figure B.2: The arrival time of the current sheet at the $z = 0$ cm array is shown. This time represents final velocity of current, the current forms at $z = -75$ cm at 0 μs on all of the pulses. Earlier gas puff timings slow the current sheet.

in Fig. B.4. Although this seems scattered, there are trends which guide the operation of the machine. The first being, the timings with the longest quiescent periods do not generate the most repeatable pulses. The settings with the longest quiescent periods often generate Z-pinches with two different behaviors of the mode amplitudes. In one type of pulse, the mode amplitudes are similar to Fig. 6.9. In the second type of pulse, the mode amplitude does not change character until the current has significantly decreased, similar to

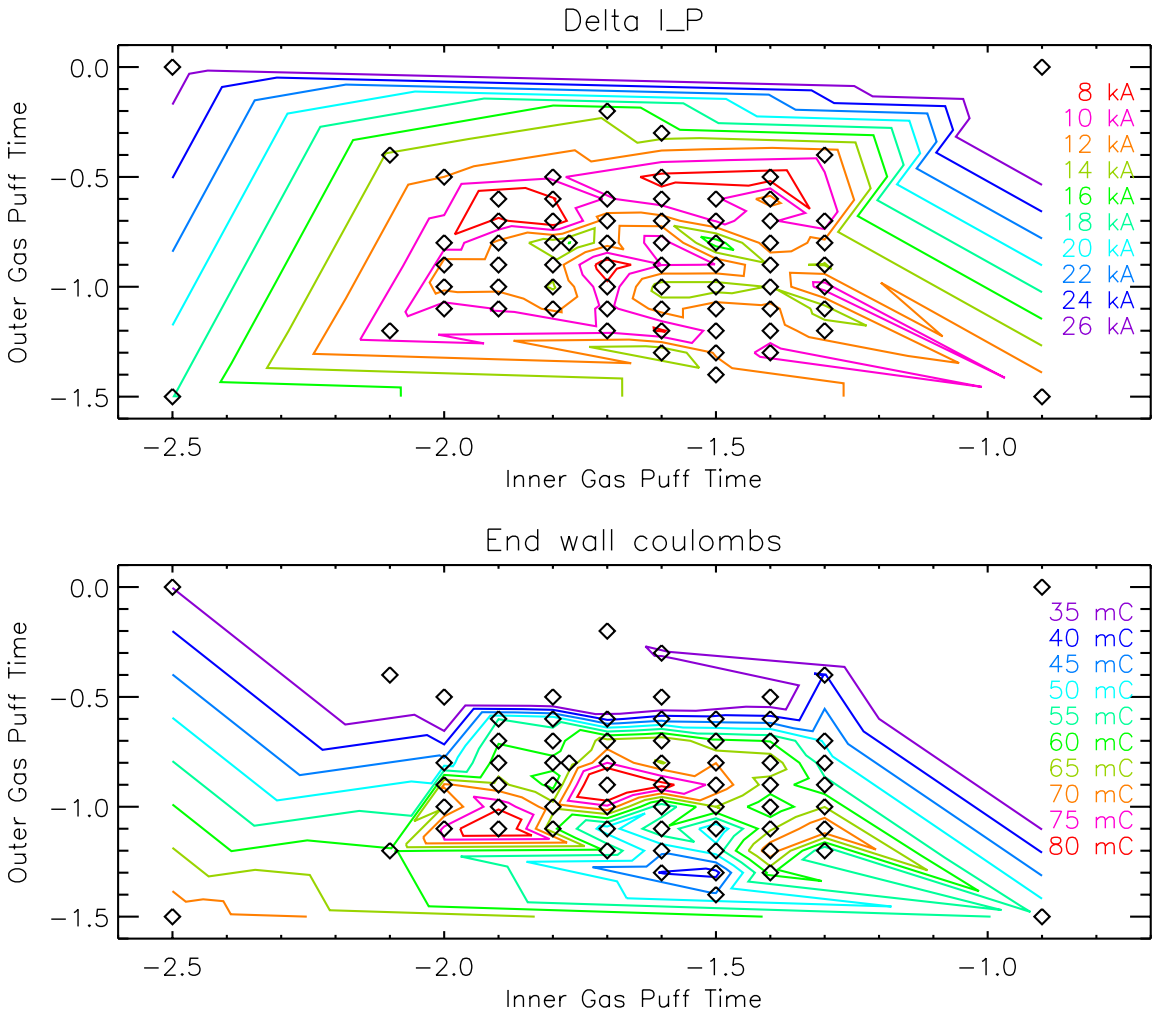

Figure B.3: Shown are contours of the repeatability of the plasma current and the total number of coulombs exiting the end during the first half cycle of the current. Desirable characteristics are shown in red. The reproducibility of the driving current indicates timings where the Z-pinches have a similar behavior. The total coulombs exiting the endwall are measured with a Rogowski coil. It has been correlated with the length of the quiescent period.

Fig. 6.33. When the pulses are not reproducible, surveys which rely on multiple pulses may give ambiguous results. The experiment often uses gas pulse settings which do not create the longest quiescent periods for this reason.

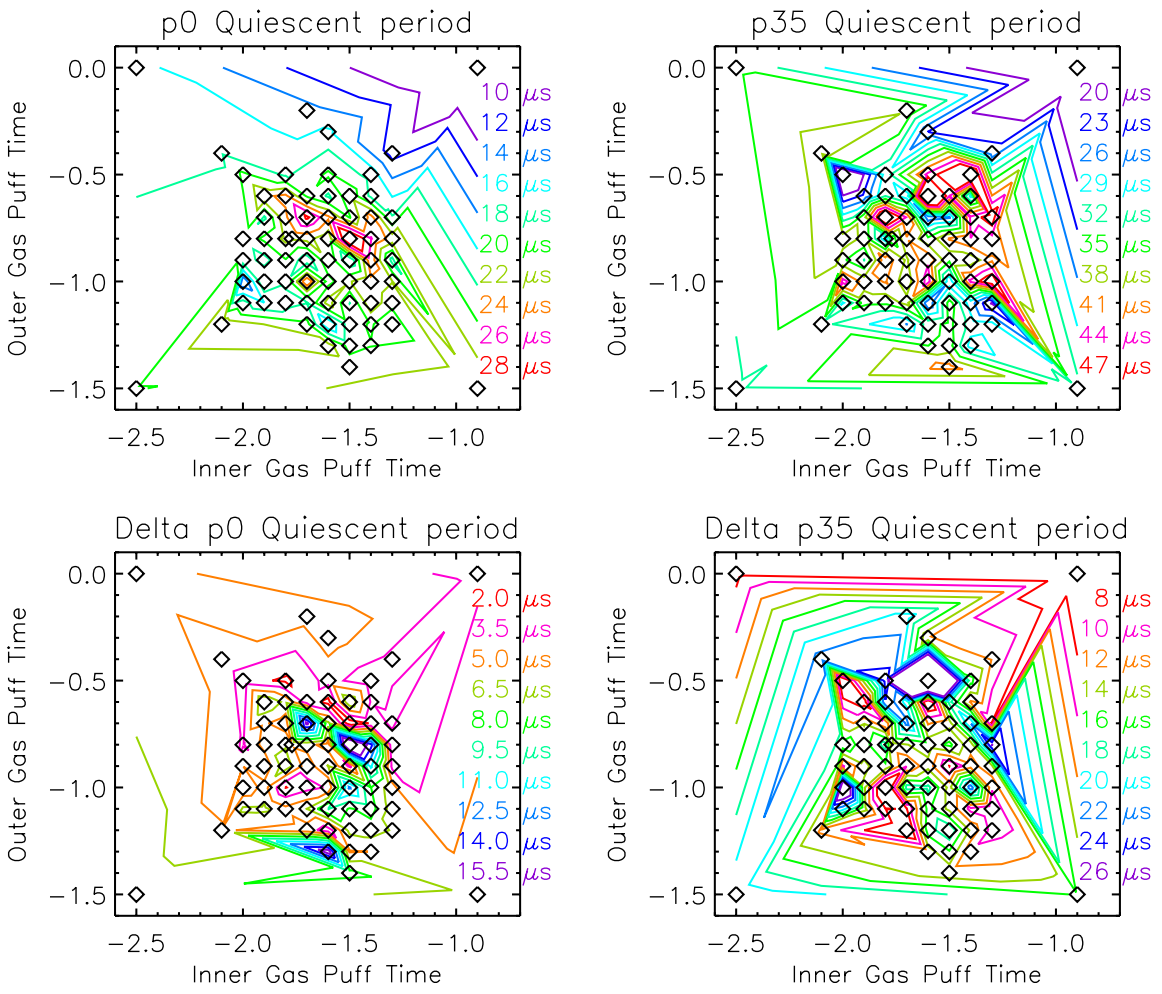

Figure B.4: Shown are the average quiescent periods at the $z = 0$ and $z = 35$ arrays. The standard deviations of the quiescent periods are also shown. The red contours indicate longer quiescent periods and more repeatable pulses.

Appendix C

IDL CODES

The large amounts of data and pulses on the ZaP experiment require analysis codes. The codes included in this chapter are used to reduce the data in the trees or to calculate theoretic predictions. All the codes are written in IDL because MDSplus can automatically run IDL codes during a pulse. Most of the subroutines needed for the codes are included or may be written similar to the included samples. The functions which are not included in this work, i.e. *data*, *stremo*, etc., are presently stored in a repository at the University of Washington. These functions are specific to reading the data from the trees or are used for plots of the data. They may be obtained by contacting the author or the Plasma Dynamics Group at the University of Washington. Since the codes run without user intervention, the results from them should always be verified. Each section in this appendix describes the codes and the methods to verify the analyzed data.

C.1 Snowplow Codes

The codes in this section are used to calculate the location of the current sheet, the current and the shape of snowplow. The theoretical predictions are described in Sec. 3.2. *snow_plow_time.pro* calculates the location of the current sheet and compares it to the theoretical prediction. The shape of the snowplow is calculated with *con_cur_sp_dis.pro*.

C.1.1 snow_plow_time.pro

```
pro snow_plow_time,shot,ps=ps,V_mb=V_mb
default,shot,20718030
default,ps,0
default,v_mb,9
if ps eq 1 then begin
  SET_PLOT, 'PS'
  DEVICE, FILENAME='snow_plow_'+stremo(shot)+'.ps', /color, $
         /portrait,ysize=10,yoffset=1.2
  !p.thick=3
  !x.thick=3
  !y.thick=3
  !p.charthick=2
  !p.charsize=1
  !p.multi=0
endif
n_el=321
dela_t=2.17e-3
P=3.54E-01
gamma=1.4
R0=0.01
C0=0.00068
L0=0.000000543
V0=v_mb*1000.
k=1
mu0=4*!PI*0.0000001
n_total=5e19
r_inner=0.05
r_outer=0.1
```

```
l_inner=1.0
m=2*1.67E-27*4.5
bb=2*(r_outer-r_inner)/r_outer
rho_0=n_total*m/(l_inner*!PI*(r_outer^2-r_inner^2))
P_calc=R0*sqrt(C0/L0)
q_calc=(C0^2*V0^2*bb^2/(2*!PI*r_inner^2*rho_0*L0^2*k))*(mu0/(4*!PI))^3
q=q_calc

tau=findgen(n_el)*dela_t
a=fltarr(n_el)+(3.+gamma)
b=fltarr(n_el)
c=fltarr(n_el)
phi=fltarr(n_el)
y=fltarr(n_el)
phi(1)=dela_t
for i=2,n_el-1 do begin
   b(i)=-2*(gamma-1)*y(i-2)
   c(i)=-8*y(i-1)^2+(3+gamma)*y(i-2)^2-8*dela_t^2*q*phi(i-1)^2
   y(i)=(-b(i)+SQRT(b(i)^2-4*a(i)*c(i)))/(2*a(i))
   phi(i)=(4*(1+y(i-1))*phi(i-1)-2*(1+y(i-2))*phi(i-2)+ $
           P*dela_t*phi(i-2)-2*dela_t^2*phi(i-1))/ $
           (2*(1+y(i))+dela_t*P)
endfor
dy_dt=fltarr(n_el)
d2y_dt2=fltarr(n_el)
eqn1=fltarr(n_el)
eqn1a=fltarr(n_el)
eqn1b=fltarr(n_el)
eqn2=fltarr(n_el)
for i=1,n_el-2 do begin
```

```
   dy_dt(i)=(tau(i+1)-tau(i-1))/(2*dela_t)
   d2y_dt2(i)=(tau(i-1)^2-2*tau(i)^2+tau(i+1)^2)/dela_t^2
   eqn1(i)=d2y_dt2(i)/2+(gamma-1)*dy_dt(i)^2/2-q*y(i)^2
   eqn1a(i)=((1+y(i))*phi(i)-2*(1+y(i))*phi(i)+ $
            (1+y(i-1))*phi(i-1))/dela_t^2
   eqn1b(i)=(phi(i+1)-phi(i-1))/(2*dela_t)
   eqn2(i)=eqn1a(i)+P*eqn1b(i)+phi(i)
endfor

t=tau*SQRT(L0*C0)
loc=y*(mu0*bb/(4*!PI*L0))^(-1)
cur=phi*V0*C0/sqrt(L0*C0)
;get shot data
ss,shot
ip=data('sigadd(\i_rf_ign,\i_lf_ign)',xaxis=t_ip)
b_name=['\b_n120','\b_n100','\b_n80','\b_n70', $
          '\b_n65','\b_n60','\b_n55','\b_n45','\b_n40','\b_n35', $
          '\b_n30','\b_n25','\b_n20','\b_n15','\b_n10','\b_n5','\b_p0', $
          '\b_p5','\b_p10','\b_p15','\b_p20']
n_probe=n_elements(b_name)
n_time=n_elements(t_ip)
b_all=fltarr(n_time,n_probe)
arrive=fltarr(n_probe)
z_probe=findgen(n_probe)*5.-90
z_probe(0)=-120
z_probe(1)=-100
z_probe(where(z_probe ge -75))=z_probe(where(z_probe ge -75))+5
z_probe(where(z_probe ge -50))=z_probe(where(z_probe ge -50))+5
for i=0,n_probe-1 do begin
   b_temp=data(b_name(i),xaxis=t_temp)
```

```
   b_all(*,i)=interpol(b_temp,t_temp,t_ip)
   arrive(i)=t_ip(min(where((dsmooth(b_all(*,i),3,5) ge $
             .03*max(dsmooth(b_all(*,0),3,5))) and $
             (t_ip ge arrive((i-1)>0)))))
endfor
!p.multi=[0,1,2]
plot,t*1e6,loc*100-75, $
     ytitle='Z (cm)', $
     title='Varing Current, Snowplow Model'
oplot,(arrive-2.5e-6)*1e6,z_probe,co=2
plot,t*1e6,cur*1e-3, $
     ytitle='I (kA)', $
     xtitle='Time (!4l!3s)'
oplot,(t_ip-2.5e-6)*1e6,ip*1e-3,co=2
xyouts,.7,.04,'Pulse='+stremo(shot),/normal
xyouts,.8,.84,'Theory',/normal
xyouts,.8,.81,'Experimental',/normal,co=2
!p.region=0
if ps eq 1 then begin
  device,/close
  SET_PLOT, 'win'
  !p.thick=1
  !x.thick=1
  !y.thick=1
  !p.charthick=1
  !p.charsize=1.5
  !p.multi=0
endif
!p.multi=0
end
```

C.1.2 con_cur_sp_dis.pro

```
pro con_cur_sp_dis
ps=1
if ps eq 1 then begin
  SET_PLOT, 'PS'
  DEVICE, FILENAME='con_cur_sp_dis.ps', /color, $
         /portrait,ysize=10,yoffset=1.2
  !p.thick=3
  !x.thick=3
  !y.thick=3
  !p.charthick=2
  !p.charsize=1.2
  !p.multi=0
endif

default,uc,0.
default,gamma,5./3.   ;Ideal MHD
!p.multi=[0,1,3]
for ii=0,0 do begin

uc=ii*.2
uc=.1
ds=.001
pi=!dpi
npts=1.6/ds+1
s=dindgen(npts)*ds

r_anal=SQRT(1+2*s)
z_anal=(1/2.)*(SQRT(s*(2.+4.*s))- $
```

```
              alog(sqrt(2*s)+SQRT(1+2*s)))

sig_anal=2*s^2/(2.*z_anal>ds)
V_anal=2.*z_anal/(2*s>ds)

plot,-z_anal,r_anal,yra=[1,2],xra=[-1,0.2], $
     ytitle='r!dn!n', $;xtitle='z!dn!n',
     title='Snowplow Shape'

;initializing arrays
r=dblarr(npts)+1.
z=dblarr(npts)
th=dblarr(npts)
sig=dblarr(npts)
V=dblarr(npts)
eps=dblarr(npts)
delta=dblarr(npts)
P=dblarr(npts)

;initial point
a_V=sqrt(8./3.+(1-uc^2)*(gamma-1)/(3.*gamma))
a_e=(1.-uc^2)/a_V
a_s=1./a_V
a_t=2.*a_s

sqrt_ds=sqrt(ds)
r(1)=(1+ds)
th(1)=a_t*sqrt_ds
z(1)=ds*sin(th(1))
sig(1)=a_s*sqrt_ds
```

```
V(1)=a_V*sqrt_ds
eps(1)=a_e*sqrt_ds
P(1)=cos(th(1))^2+1./(r(1)^2)
delta(1)=((gamma-1)/gamma)*(eps(1)/(r(1)*P(1)))

;runge-kutta method
for i=long(1),npts-2 do begin
;first
r_t1=r(i)
z_t1=z(i)
th_t1=th(i)
sig_t1=sig(i)
V_t1=V(i)
eps_t1=eps(i)
P_t1=P(i)
delta_t1=delta(i)

sigV_t1=sig_t1*V_t1
sigV2_t1=sig_t1*V_t1^2
epsV_t1=eps_t1*V_t1

k1_t=ds*(r_t1/(sig_t1*V_t1^2))*(cos(th_t1)^2-1/(r_t1^2))
k1_sv=ds*(r_t1*cos(th_t1))
dp_ds=-2*cos(th_t1)*(1/(r_t1^3)+sin(th_t1)*k1_t/ds)
k1_sv2=ds*(r_t1*cos(th_t1)*sin(th_t1)-(r_t1*delta_t1/2.) * dp_ds)
k1_ev=ds*(r_t1*cos(th_t1)*(cos(th_t1)^2+ $
       (sin(th_t1)-V_t1)^2-uc^2)+r_t1*delta_t1*dp_ds)
k1_r=ds*cos(th_t1)
k1_z=ds*sin(th_t1)
```

```
sigV_t2=sigV_t1+k1_sv/2.
sigV2_t2=sigV2_t1+k1_sv2/2.
epsV_t2=epsV_t1+k1_ev/2.
r_t2=r_t1+k1_r/2.
z_t2=z_t1+k1_z/2.
th_t2=th_t1+k1_t/2.

sig_t2=sigV_t2^2/sigV2_t2
V_t2=sigV2_t2/sigV_t2
eps_t2=epsV_t2/V_t2
P_t2=cos(th_t2)^2+1./(r_t2^2)
delta_t2=((gamma-1)/gamma)*(eps_t2/(r_t2*P_t2))

;second
k2_t=ds*(r_t2/(sig_t2*V_t2^2))*(cos(th_t2)^2-1/(r_t2^2))
k2_sv=ds*(r_t2*cos(th_t2))
dp_ds=-2*cos(th_t2)*(1/(r_t2^3)+sin(th_t2)*k2_t/ds)
k2_sv2=ds*(r_t2*cos(th_t2)*sin(th_t2)-(r_t2*delta_t2/2.) * dp_ds)
k2_ev=ds*(r_t2*cos(th_t2)*(cos(th_t2)^2+ $
       (sin(th_t2)-V_t2)^2-uc^2)+r_t2*delta_t2*dp_ds)
k2_r=ds*cos(th_t2)
k2_z=ds*sin(th_t2)

sigV_t3=sigV_t1+k2_sv/2.
sigV2_t3=sigV2_t1+k2_sv2/2.
epsV_t3=epsV_t1+k2_ev/2.
r_t3=r_t1+k2_r/2.
z_t3=z_t1+k2_z/2.
th_t3=th_t1+k2_t/2.
```

```
sig_t3=sigV_t3^2/sigV2_t3
V_t3=sigV2_t3/sigV_t3
eps_t3=epsV_t3/V_t3
P_t3=cos(th_t3)^2+1./(r_t3^2)
delta_t3=((gamma-1)/gamma)*(eps_t3/(r_t3*P_t3))

;third
k3_t=ds*(r_t3/(sig_t3*V_t3^2))*(cos(th_t3)^2-1/(r_t3^2))
k3_sv=ds*(r_t3*cos(th_t3))
dp_ds=-2*cos(th_t3)*(1/(r_t3^3)+sin(th_t3)*k3_t/ds)
k3_sv2=ds*(r_t3*cos(th_t3)*sin(th_t3)-(r_t3*delta_t3/2.) * dp_ds)
k3_ev=ds*(r_t3*cos(th_t3)*(cos(th_t3)^2+ $
       (sin(th_t3)-V_t3)^2-uc^2)+r_t3*delta_t3*dp_ds)
k3_r=ds*cos(th_t3)
k3_z=ds*sin(th_t3)

sigV_t4=sigV_t1+k3_sv
sigV2_t4=sigV2_t1+k3_sv2
epsV_t4=epsV_t1+k3_ev
r_t4=r_t1+k3_r
z_t4=z_t1+k3_z
th_t4=th_t1+k3_t

sig_t4=sigV_t4^2/sigV2_t4
V_t4=sigV2_t4/sigV_t4
eps_t4=epsV_t4/V_t4
P_t4=cos(th_t4)^2+1./(r_t4^2)
delta_t4=((gamma-1)/gamma)*(eps_t4/(r_t4*P_t4))

;fourth
```

```
k4_t=ds*(r_t4/(sig_t4*V_t4^2))*(cos(th_t4)^2-1/(r_t4^2))
k4_sv=ds*(r_t4*cos(th_t4))
dp_ds=-2*cos(th_t4)*(1/(r_t4^3)+sin(th_t4)*k4_t/ds)
k4_sv2=ds*(r_t4*cos(th_t4)*sin(th_t4)-(r_t4*delta_t4/2.) * dp_ds)
k4_ev=ds*(r_t4*cos(th_t4)*(cos(th_t4)^2+ $
      (sin(th_t4)-V_t4)^2-uc^2)+r_t4*delta_t4*dp_ds)
k4_r=ds*cos(th_t4)
k4_z=ds*sin(th_t4)

sigV_t=sigV_t1+(k1_sv+2*k2_sv+2*k3_sv+k4_sv)/6.
sigV2_t=sigV2_t1+(k1_sv2+2*k2_sv2+2*k3_sv2+k4_sv2)/6.
epsV_t=epsV_t1+(k1_ev+2*k2_ev+2*k3_ev+k4_ev)/6.
r(i+1)=r_t1+(k1_r+2*k2_r+2*k3_r+k4_r)/6.
z(i+1)=z_t1+(k1_z+2*k2_z+2*k3_z+k4_z)/6.
th(i+1)=th_t1+(k1_t+2*k2_t+2*k3_t+k4_t)/6.

sig(i+1)=sigV_t^2/sigV2_t
V(i+1)=sigV2_t/sigV_t
eps(i+1)=epsV_t/V(i+1)
P(i+1)=cos(th(i+1))^2+1./(r(i+1)^2)
delta(i+1)=((gamma-1)/gamma)*(eps(i+1)/(r(i+1)*P(i+1)))

endfor
oplot,-z,r,co=2
oplot,-z+.5*delta,r,co=4
oplot,-z-.5*delta,r,co=4
oplot,-z_anal,r_anal
plot,-z_anal,V_anal,xra=[-1,0.2],yra=[0,.8], $
     ytitle='V!dn!n', $;xtitle='z!dn!n',
     title='Snowplow Tangential Velocity'
```

```
oplot,-z,V,co=2
plot,-z_anal,sig_anal,xra=[-1,0.2], $
     xtitle='z!dn!n',ytitle='!4r!3!dn!n', $
     title='Snowplow Density'
oplot,-z,sig,co=2
endfor
print,delta(0:10)

!p.region=0
if ps eq 1 then begin
  device,/close
  SET_PLOT, 'win'
  !p.thick=1
  !x.thick=1
  !y.thick=1
  !p.charthick=1
  !p.charsize=1.5
  !p.multi=0
endif
!p.multi=0
end
```

C.2 Magnetic Probe Codes

The magnetic probes have been described in Sec. 5.5. The calibration factors for the magnetic probes are found with *cal_axial_b_cal_fact_11214.pro*. This code finds a scale factor between the maximum probe signal and the maximum magnetic field the current would produce at the wall. *rundown_vel.pro* is used to find the location velocity and mass of the current sheet. The method to calculate these parameters were discussed in Sec. 5.5.5. The axial position of the current sheet is not correctly found when a filament is ahead of the current sheet. The current sheet velocity from pulses with large negative initial magnetic fields should be compared with similar to pulses. The pseudoinverse for the magnetic mode calculation is found with *find_mode_array.pro*. The angle of each probe is entered into *phi*. The pseudoinverse for various maximum m's is then printed to the screen. The calculations can be checked by using evenly spaced probes and ensuring that the matrix is diagonal. *mode_calculation.pro* is used to calculate the Fourier components of the magnetic fields. The pseudoinverse for the given probe geometry is then entered into *mode_calculation.pro* to save time. The iterative calculation of the wall magnetic fields is made with *find_b_wall.pro*. The method used to calculate the fields is described in Sec. A.3.2. The code can take over 24 hours to run on a PC.

C.2.1 cal_axial_b_cal_fact_11214.pro

```
pro cal_axial_b_cal_fact_11214,xra=xra, $
        cal_fact_fast=cal_fact_fast, $
        cal_fact_slow=cal_fact_slow, $
        ps=ps

default,ps,0
mydevice = !D.NAME
if ps eq 1 then begin
  charsize=.5
  SET_PLOT, 'PS'
```

```
  DEVICE, FILENAME='cal_fact_11214.ps', /portrait, /color, $
          ysize=25,yoffset=1.2
  !p.multi=[0,2,3]
endif

default,xra,[-0.2e-3,0.8e-3]
t_end=2e-4

b_name=['n120','n100','n80','n75','n70','n65','n60','n55', $
         'n50', 'n45','n40','n35', n30','n20','n15','n10', $
          'n5',  'p5','p10','p15','p20']
n_probe=n_elements(b_name)

fast_clock_data=[[11214009,11214001,4], $
                 [11214010,11214001,4], $
                 [11214019,11214021,2], $
                 [11214020,11214021,2], $
                 [11214026,11214022,1], $
                 [11214027,11214022,1]]

slow_clock_data=[[11214011,11214012,4], $
                 [11214013,11214012,4], $
                 [11214017,11214014,2], $
                 [11214018,11214014,2], $
                 [11214029,11214028,1], $
                 [11214030,11214028,1]]
fast_clock_shot=transpose(fast_clock_data(0,*))
fast_clock_base=transpose(fast_clock_data(1,*))
fast_clock_cap=transpose(fast_clock_data(2,*))
n_shot_fast=n_elements(fast_clock_shot)
```

```
slow_clock_shot=transpose(slow_clock_data(0,*))
slow_clock_base=transpose(slow_clock_data(1,*))
slow_clock_cap=transpose(slow_clock_data(2,*))
n_shot_slow=n_elements(slow_clock_shot)

cal_fact_fast=fltarr(n_probe,n_shot_fast)
cal_fact_fa_sm=fltarr(n_probe,n_shot_fast)
cal_fact_slow=fltarr(n_probe,n_shot_slow)

for j=0,n_shot_fast-1 do begin
   ss,fast_clock_shot(j)
   r_wall=data('\r_wall')
   i_p=data('sub_baseline_string("\\i_lf_ign_l", $
            '+stremo(fast_clock_base(j))+')',xaxis=time_i)
   i_p=i_p*(2e-7)/r_wall
   i_p_sm=dsmooth(i_p)
   max_i_p=max(abs(i_p))
   max_i_p_sm=max(abs(i_p_sm))
   plot,time_i,i_p_sm,xra=xra, $
        title='shot='+stremo(fast_clock_shot(j))+' fast'
   for i=0,n_probe-1 do begin
      probe_raw=data('sub_baseline_string( $
                     "\\b_'+b_name(i)+':raw", $
                     '+stremo(fast_clock_base(j))+')', $
                     xaxis=time_probe)
      n_raw=n_elements(probe_raw)
      n_0=min(where(time_probe ge 0))
      n_end=min(where(time_probe ge (t_end < $
                                     time_probe(n_raw-1))))
```

```
      n_max=n_0+min(where(abs(probe_raw(n_0:n_end)) eq $
                          max(abs(probe_raw(n_0:n_end)))))
      probe_raw_sm=dsmooth(probe_raw)
      n_max_sm=n_0+min(where(abs(probe_raw_sm(n_0:n_end)) eq $
                             max(abs(probe_raw_sm(n_0:n_end)))))
      cal_fact_fast(i,j)=max_i_p/probe_raw(n_max)
      cal_fact_fa_sm(i,j)=max_i_p_sm/probe_raw(n_max_sm)
      oplot,time_probe,probe_raw_sm*cal_fact_fa_sm(i,j),color=2+i
      if ps ne 1 then $
        xyouts,.8,.9-.02*i,b_name(i),/normal,color=2+i $
        else xyouts,.5,.9-.03*i,b_name(i),/normal,color=2+i
    endfor
endfor

for j=0,n_shot_slow-1 do begin
   ss,slow_clock_shot(j)
   r_wall=data('\r_wall')
   i_p=data('sub_baseline_string("\\i_lf_ign_l", $
            '+stremo(slow_clock_base(j))+')',xaxis=time_i)
   i_p=i_p*(2e-7)/r_wall
   max_i_p=max(abs(i_p))
   plot,time_i,i_p,xra=xra, $
        title='shot='+stremo(slow_clock_shot(j))+' slow'
   for i=0,n_probe-1 do begin
      probe_raw=data('sub_baseline_string( $
                     "\\b_'+b_name(i)+':raw", $
                     '+stremo(slow_clock_base(j))+')', $
                      xaxis=time_probe)
      n_raw=n_elements(probe_raw)
      n_0=min(where(time_probe ge 0))
```

```
      n_end=min(where(time_probe ge (t_end < $
                                    time_probe(n_raw-1))))
      n_max=n_0+min(where(abs(probe_raw(n_0:n_end)) eq  $
                          max(abs(probe_raw(n_0:n_end)))))
      cal_fact_slow(i,j)=max_i_p/probe_raw(n_max)
      oplot,time_probe,probe_raw*cal_fact_slow(i,j),color=2+i
      if ps ne 1 then $
        xyouts,.8,.9-.02*i,b_name(i),/normal,color=2+i $
        else xyouts,.5,.9-.03*i,b_name(i),/normal,color=2+i
      endfor
endfor
print,'slow'
print,cal_fact_slow
print,'fast'
print,cal_fact_fast
print,'smooth'
print,cal_fact_fa_sm

cal_fact_ave=fltarr(n_probe)
cal_fact_sm=fltarr(n_probe)
err_cal_fact_ave=fltarr(n_probe)
err_cal_fact_sm=fltarr(n_probe)

for i=0,n_probe-1 do begin
   cal_fact_ave(i)=total(cal_fact_fast(i,*))/n_shot_fast
   cal_fact_sm(i)=total(cal_fact_fa_sm(i,*))/n_shot_fast
   err_cal_fact_ave(i)=total((cal_fact_fast(i,*)- $
                              cal_fact_ave(i))^2)
   err_cal_fact_sm(i)=total((cal_fact_fa_sm(i,*)- $
                             cal_fact_sm(i))^2)
```

```
endfor
print,cal_fact_ave,sqrt(err_cal_fact_ave)/ $
                       (cal_fact_ave*sqrt(n_shot_fast))
print,cal_fact_sm,sqrt(err_cal_fact_sm)/ $
                      (cal_fact_sm*sqrt(n_shot_fast))

if ps eq 1 then begin
  DEVICE, /CLOSE
  SET_PLOT, mydevice
  !p.multi=0
endif
end
```

C.2.2 rundown_vel.pro

```
pro rundown_vel,shot,vel_65_25=vel_65_25,vel_65_45=vel_65_45, $
                vel_45_25=vel_45_25, $
                t_rundown=t_rundown,v_rundown=v_rundown, $
                mass_rundown=mass_rundown, $
                t2_rundown=t2_rundown,v2_rundown=v2_rundown, $
                mass2_rundown=mass2_rundown,ps=ps,p_ps=p_ps

default,shot,21205036
default,ps,0
default,p_ps,0
ss,shot

b_n120=data('\b_n120',xaxis=time_120)
b_n65=data('\b_n65',xaxis=time_65)
b_n55=data('\b_n55',xaxis=time_55)
b_n45=data('\b_n45',xaxis=time_45)
b_n35=data('\b_n35',xaxis=time_35)
b_n25=data('\b_n25',xaxis=time_25)

sm_b_n120=dsmooth(b_n120,17,19)
sm_b_n65=dsmooth(b_n65,17,19)
sm_b_n55=dsmooth(b_n55,17,19)
sm_b_n45=dsmooth(b_n45,17,19)
sm_b_n35=dsmooth(b_n35,17,19)
sm_b_n25=dsmooth(b_n25,17,19)
arrive_n120=time_120(min(where $
            ((sm_b_n120 ge .05*max(sm_b_n120)) and  $
             (time_120 ge 0))))
```

```
arrive_n65=time_65(min(where $
            ((sm_b_n65 ge .05*max(sm_b_n65)) and  $
             (time_65 ge arrive_n120+.05e-6))))
arrive_n55=time_55(min(where $
            ((sm_b_n55 ge .05*max(sm_b_n55)) and  $
             (time_55 ge arrive_n65+.05e-6))))
arrive_n45=time_45(min(where $
            ((sm_b_n45 ge .05*max(sm_b_n45)) and  $
             (time_45 ge arrive_n55+.05e-6))))
arrive_n35=time_35(min(where $
            ((sm_b_n35 ge .05*max(sm_b_n35)) and  $
             (time_35 ge arrive_n45+.05e-6))))
arrive_n25=time_25(min(where $
            ((sm_b_n25 ge .05*max(sm_b_n25)) and  $
             (time_25 ge arrive_n35+.05e-6))))

sm2_b_n65=interpol(sm_b_n65,time_65,time_120)
sm2_b_n55=interpol(sm_b_n55,time_55,time_120)
sm2_b_n45=interpol(sm_b_n45,time_45,time_120)
sm2_b_n35=interpol(sm_b_n35,time_35,time_120)
sm2_b_n25=interpol(sm_b_n25,time_25,time_120)
arrive2_n120=time_120(min(where $
            ((sm_b_n120 ge .05*max(sm_b_n120)) and  $
             (time_120 ge 0))))
arrive2_n65=time_120(min(where $
             ((sm2_b_n65 ge .3*sm_b_n120) and  $
              (time_120 ge arrive2_n120+.05e-6))))
arrive2_n55=time_120(min(where $
             ((sm2_b_n55 ge .3*sm_b_n120) and  $
              (time_120 ge arrive2_n65+.05e-6))))
```

```
arrive2_n45=time_120(min(where $
             ((sm2_b_n45 ge .3*sm_b_n120) and  $
              (time_120 ge arrive2_n55+.05e-6))))
arrive2_n35=time_120(min(where $
             ((sm2_b_n35 ge .3*sm_b_n120) and  $
              (time_120 ge arrive2_n45+.05e-6))))
arrive2_n25=time_120(min(where $
             ((sm2_b_n25 ge .3*sm_b_n120) and  $
              (time_120 ge arrive2_n35+.05e-6))))

vel_65_25=.4/(arrive_n25-arrive_n65)
vel_65_45=.2/(arrive_n45-arrive_n65)
vel_45_25=.2/(arrive_n25-arrive_n45)

v_rundown=fltarr(5)
v_rundown(1)=.1/(arrive_n55-arrive_n65)
v_rundown(2)=.1/(arrive_n45-arrive_n55)
v_rundown(3)=.1/(arrive_n35-arrive_n45)
v_rundown(4)=.1/(arrive_n25-arrive_n35)
t_rundown=fltarr(5)
t_rundown(0)=arrive_n120
t_rundown(1)=(arrive_n55+arrive_n65)/2.
t_rundown(2)=(arrive_n45+arrive_n55)/2.
t_rundown(3)=(arrive_n35+arrive_n45)/2.
t_rundown(4)=(arrive_n25+arrive_n35)/2.
v2_rundown=fltarr(5)
v2_rundown(1)=.1/(arrive2_n55-arrive2_n65)
v2_rundown(2)=.1/(arrive2_n45-arrive2_n55)
v2_rundown(3)=.1/(arrive2_n35-arrive2_n45)
v2_rundown(4)=.1/(arrive2_n25-arrive2_n35)
```

```
t2_rundown=fltarr(5)
t2_rundown(0)=arrive_n120
t2_rundown(1)=(arrive2_n55+arrive2_n65)/2.
t2_rundown(2)=(arrive2_n45+arrive2_n55)/2.
t2_rundown(3)=(arrive2_n35+arrive2_n45)/2.
t2_rundown(4)=(arrive2_n25+arrive2_n35)/2.

n_b120_rundown=fltarr(5)
for i=0,4 do $
   n_b120_rundown(i)=min(where(time_120 ge t_rundown(i)))
n2_b120_rundown=fltarr(5)
for i=0,4 do $
   n2_b120_rundown(i)=min(where(time_120 ge t2_rundown(i)))

pi=!dpi
r_wall=.0963
r_inner=.0508
mu_0=pi*4e-7
F_rundown=(pi*r_wall^2*b_n120^2/mu_0)*alog(r_wall/r_inner)

impulse=fltarr(5)
for i=1,4 do $
   impulse(i)=impulse(i-1)+ $
    int_tabulated( $
     time_120(n_b120_rundown(i-1):n_b120_rundown(i)), $
      F_rundown(n_b120_rundown(i-1):n_b120_rundown(i)))
impulse2=fltarr(5)
for i=1,4 do $
   impulse2(i)=impulse2(i-1)+ $
    int_tabulated( $
```

```
     time_120(n2_b120_rundown(i-1):n2_b120_rundown(i)), $
      F_rundown(n2_b120_rundown(i-1):n2_b120_rundown(i)))

mass_rundown=impulse/(v_rundown>1.)
mass2_rundown=impulse2/(v2_rundown>1.)

if ps ne 1 then begin
if p_ps eq 1 then begin
  SET_PLOT, 'PS'
  DEVICE, FILENAME='rundown_vel_'+stremo(shot)+'.ps', $
          SET_FONT='Times', /color , $
          /portrait,ysize=25,yoffset=1.2
  !p.thick=3
  !x.thick=3
  !y.thick=3
  !p.charthick=2
  !p.charsize=1
endif
!p.multi=[0,3,3]
plot,time_65*1e6,b_n65,xra=[0,50],yra=[0,max(sm_b_n120)], $
     title='Magnetic Feild Shot='+stremo(shot)
oplot,time_65*1e6,sm_b_n65,co=4
vline,arrive_n65*1e6
oplot,time_45*1e6,b_n45
oplot,time_45*1e6,sm_b_n45,co=4
vline,arrive_n45*1e6
oplot,time_25*1e6,b_n25
oplot,time_25*1e6,sm_b_n25,co=4
vline,arrive_n25*1e6
oplot,time_55*1e6,b_n55
```

```
oplot,time_55*1e6,sm_b_n55,co=4
vline,arrive_n55*1e6
oplot,time_35*1e6,b_n35
oplot,time_35*1e6,sm_b_n35,co=4
vline,arrive_n35*1e6
oplot,time_120*1e6,b_n120
oplot,time_120*1e6,sm_b_n120,co=4
vline,arrive_n120*1e6
vline,arrive2_n65*1e6,co=2
vline,arrive2_n55*1e6,co=2
vline,arrive2_n45*1e6,co=2
vline,arrive2_n35*1e6,co=2
vline,arrive2_n25*1e6,co=2

xra=[0,arrive2_n25+5e-6]
plot,t_rundown,v_rundown,xra=xra,title='Velocity'
oplot,t2_rundown,v2_rundown,co=2

plot,time_120,F_rundown,xra=xra,title='Force'

plot,t_rundown,impulse,xra=xra,yra=[0,impulse2(4)], $
     title='Impulse'
oplot,t2_rundown,impulse2,co=2
plot,t_rundown,mass_rundown,xra=xra, $
     yra=[0,max([mass_rundown,mass2_rundown])], $
     title='Total mass'
oplot,t2_rundown,mass2_rundown,co=2
plot,t_rundown,mass_rundown/1.6726e-27,xra=xra, $
     yra=[0,max([mass_rundown,mass2_rundown])/1.6726e-27], $
     title='Total ions'
```

```
oplot,t2_rundown,mass2_rundown/1.6726e-27,co=2
plot,t_rundown,mass_rundown/1.6726e-27,xra=xra, $
     yra=[0,max([mass_rundown,mass2_rundown])/1.6726e-27], $
     title='Total ions'
oplot,t2_rundown,mass2_rundown/1.6726e-27,co=2
vline,arrive_n25
vline,arrive2_n25,co=2
!p.multi=0
if p_ps eq 1 then begin
  device,/close
  SET_PLOT, 'win'
  !p.thick=1
  !x.thick=1
  !y.thick=1
  !p.charthick=1
  !p.charsize=1.5
  !p.multi=0
endif
endif else begin
xra=[0,arrive2_n25+5e-6]*1e6

plot,t_rundown*1e6,mass_rundown/1.6726e-27,xra=xra, $
     yra=[0,max([mass_rundown,mass2_rundown])/1.6726e-27], $
     title='Total ions'
oplot,t2_rundown*1e6,mass2_rundown/1.6726e-27,co=2
vline,arrive_n25*1e6
vline,arrive2_n25*1e6,co=2
endelse
end
```

C.2.3 find_mode_array.pro

```
;       This code calculates the inversion array needed to
; find the amplitudes and phases of the mode amplitudes when
; the probes are not equally spaced. The famous missing probe
; problem

pro find_mode_array

;azimuthal position of said probes
phi =   [  0.0,  45.0,  90.0, 135.0, $
         180.0, 225.0, 270.0, 315.0] * !dpi / 180.
phi =   [ 45.0,  90.0, 135.0, $
         180.0, 225.0, 270.0, 315.0] * !dpi / 180.

; Calculating inversion array.  This array was obtained by
; assuming f(x)=a0 + sum(a(i)*cos(i*x)) + sum(b(i)*sin(i*x)).
; To find a(or b) multiply multiply f by cos(i*x) (or sin(i*x))
; and sum over all of the probes. This will give a system of n
; equations with n unknowns from which a(i) and b(i) can by
; solved for.
nppts=n_elements(phi)
cosp=cos(phi)
sinp=sin(phi)
cos2p=cos(2*phi)
sin2p=sin(2*phi)
cos3p=cos(3*phi)
sin3p=sin(3*phi)

u = nppts
```

```
c = total( cosp )
s = total( sinp )
c2 = total( cos2p )
s2 = total( sin2p )
c3 = total( cos3p )
s3 = total( sin3p )

cc = cosp ## transpose( cosp)
sisi = sinp ## transpose( sinp)
cs = cosp ## transpose( sinp)
c2c2 = cos2p ## transpose( cos2p)
s2s2 = sin2p ## transpose( sin2p)
c2s2 = cos2p ## transpose( sin2p)
cc2 = cosp ## transpose( cos2p)
ss2 = sinp ## transpose( sin2p)
cs2 = cosp ## transpose( sin2p)
sc2 = sinp ## transpose( cos2p)
cc3 = cosp ## transpose( cos3p)
ss3 = sinp ## transpose( sin3p)
cs3 = cosp ## transpose( sin3p)
sc3 = sinp ## transpose( cos3p)
c2c3 = cos2p ## transpose( cos3p)
s2s3 = sin2p ## transpose( sin3p)
c2s3 = cos2p ## transpose( sin3p)
s2c3 = sin2p ## transpose( cos3p)
c3c3 = cos3p ## transpose( cos3p)
s3s3 = sin3p ## transpose( sin3p)
c3s3 = cos3p ## transpose( sin3p)

mat = [ [    u,    c,    s,   c2,   s2,   c3,   s3 ] , $
```

```
        [    c,   cc,   cs,  cc2,  cs2,  cc3,  cs3 ] , $
        [    s,   cs, sisi,  sc2,  ss2,  sc3,  ss3 ] , $
        [   c2,  cc2,  sc2, c2c2, c2s2, c2c3, c2s3 ] , $
        [   s2,  cs2,  ss2, c2s2, s2s2, s2c3, s2c3 ] , $
        [   c3,  cc3,  sc3, c2c3, c2s3, c3c3, c3s3 ] , $
        [   s3,  cs3,  ss3, c2s3, s2s3, c3s3, s3s3 ] ]
inv  = invert(mat)
inv(where(abs(inv) le 1e-10))=0.0
print,'inverse (up to m=3)='
print,float(inv)

mat = [ [    u,    c,    s,   c2,   s2] , $
        [    c,   cc,   cs,  cc2,  cs2] , $
        [    s,   cs, sisi,  sc2,  ss2] , $
        [   c2,  cc2,  sc2, c2c2, c2s2] , $
        [   s2,  cs2,  ss2, c2s2, s2s2] ]
inv  = invert(mat)
inv(where(abs(inv) le 1e-10))=0.0
print,'inverse (up to m=2)='
print,float(inv)

mat = [ [    u,    c,    s] , $
        [    c,   cc,   cs] , $
        [    s,   cs, sisi] ]
inv  = invert(mat)
inv(where(abs(inv) le 1e-10))=0.0
print,'inverse (up to m=1)='
print,float(inv)

end
```

C.2.4 mode_calculation.pro

```
;       This code calculates amplitudes and phases of
; the modes as functions of time. The method is required
; for this as some probes are missing, so the remaining ones
; are unevenly
; spaced.

pro mode_calculation,m_0_p0_7=m_0_p0_7,m_1_p0_7=m_1_p0_7,c0_miss=c0_miss, $
                    c1_miss=c1_miss,s1_miss=s1_miss,time=time
shotname = stremo( mdsvalue('$shotname', /quiet, status=status) )
;probe numbers used
label = [0, 45, 90, 135, 180, 225, 270, 315]

;azimuthal position of said probes

phi =   [  0.0, 45.0, 90.0, 135.0, 180.0, 225.0, 270.0, 315.0] * !pi / 180.
phi_miss = [  0.0, 45.0, 90.0, 135.0, 180.0, 270.0, 315.0] * !pi / 180.
phi_4 = [  0.0, 90.0, 180.0, 270.0] * !pi / 180.

;number of probes is NPPTS

nppts=n_elements(label)

;find number of time points to use, NTPTS

b=data('\b_p0_0',time=t)
ntpts=n_elements(t)

;make some arrays.  ARRAY contains data,
```

```
;A0,A1,A2,A3 are the respective amplitudes of the modes versus time,
;PHASE1(2,3) is the phase of n=1(2,3) mode

array = fltarr (ntpts,nppts)
a0 = fltarr(ntpts) & a1=a0 & a2=a0 & a3 = a0
phase1 = a0 & phase2 = a0 & phase3 = a0
c0 = a0 & c1 = a0 & c2 = a0 & c3= a0
s1 = a0 & s2 = a0 & s3 = a0

array_miss = fltarr (ntpts,nppts-1)
a0_miss = fltarr(ntpts) & a1_miss=a0_miss & a2_miss=a0_miss
a3_miss = a0_miss
phase1_miss = a0_miss & phase2_miss = a0_miss
phase3_miss = a0_miss
c0_miss = a0_miss & c1_miss = a0_miss & c2_miss = a0_miss
3_miss= a0_miss
s1_miss = a0_miss & s2_miss = a0_miss & s3_miss = a0_miss

array_4 = fltarr (ntpts,4)
a0_4 = fltarr(ntpts) & a1_4=a0_4
phase1_4 = a0_4
c0_4 = a0_4
c1_4 = a0_4
s1_4 = a0_4

;loop across the probes to get data

for k=0,nppts-1 do begin

        b=data('\b_p0_'+stremo(label(k)),time=t1)
```

```
        b1=interpol(b,t1,t)
        array(*,k)=b1
endfor
time=t1
array_miss(*,0:4)=array(*,0:4)
array_miss(*,5:6)=array(*,6:7)
array_4=array(*,[0,2,4,6])
; Calculating inversion array.  This array was obtained by assuming
; f(x)=a0 + sum(a(i)*cos(i*x)) + sum(b(i)*sin(i*x)).  To find a(or b)
; multiply f by cos(i*x) (or sin(i*x)) and sum over all of the probes.
; This will give a system of n equations with n unknowns from which
; a(i) and b(i) can by solved for.

cosp=cos(phi)
sinp=sin(phi)
cos2p=cos(2*phi)
sin2p=sin(2*phi)
cos3p=cos(3*phi)
sin3p=sin(3*phi)

cosp_4=cos(phi_4)
sinp_4=sin(phi_4)

cosp_miss=cos(phi_miss)
sinp_miss=sin(phi_miss)
cos2p_miss=cos(2*phi_miss)
sin2p_miss=sin(2*phi_miss)
cos3p_miss=cos(3*phi_miss)
sin3p_miss=sin(3*phi_miss)
```

```
inv = [[.125, 0, 0, 0, 0, 0, 0], $
        [0, .250, 0, 0, 0, 0, 0], $
        [0, 0, .250, 0, 0, 0, 0], $
        [0, 0, 0, .250, 0, 0, 0], $
        [0, 0, 0, 0, .250, 0, 0], $
        [0, 0, 0, 0, 0, .250, 0], $
        [0, 0, 0, 0, 0, 0, .250]]

inv_4 = [[0.25, 0.0, 0.0], $
         [0.00, 0.5, 0.0], $
         [0.00, 0.0, 0.5]]

inv_miss  =[[ 0.199074    ,-0.104757    ,-0.104757    ,-7.06634e-009 $
            , 0.129630    , 0.104757    ,-0.0589256   ],$
            [-0.104757    , 0.398148    , 0.148148    , 3.90113e-008 $
            ,-0.183324    ,-0.148148    , 0.0833333   ],$
            [-0.104757    , 0.148148    , 0.398148    , 2.00781e-008 $
            ,-0.183324    ,-0.148148    , 0.0833334   ],$
            [-1.80641e-009, 3.15726e-008, 1.26394e-008, 0.250000     $
            -5.21567e-008,-3.30633e-008,-9.52607e-009],$
            [ 0.111111    ,-0.157135    ,-0.157135    ,-6.27398e-008 $
            , 0.444444    , 0.157135    ,-6.92547e-008],$
            [ 0.0851148   ,-0.120370    ,-0.120370    ,-2.94110e-008 $
            , 0.104757    , 0.370370    ,-0.0833334   ],$
            [ -0.104757   , 0.148148    , 0.148148    , 1.63528e-008 $
            ,-0.183324    ,-0.148148    , 0.333333    ]]


;loop over times to get results
;stop
```

```
array_t=transpose(array)
array_t_miss=transpose(array_miss)
array_t_4=transpose(array_4)
for i=0,ntpts-1 do begin

        b_miss = transpose(array_t_miss(*,i))
        uy_miss = total( b_miss )
        cy_miss = cosp_miss ## (b_miss)
        sy_miss = sinp_miss ## (b_miss)
        c2y_miss = cos2p_miss ## (b_miss)
        s2y_miss = sin2p_miss ## (b_miss)
        c3y_miss = cos3p_miss ## (b_miss)
        s3y_miss = sin3p_miss ## (b_miss)

        inh_miss = [ uy_miss, cy_miss, sy_miss, c2y_miss, s2y_miss , $
                     c3y_miss , s3y_miss ]
        res_miss = inv_miss#inh_miss

        c0_miss(i) = res_miss(0)
        c1_miss(i) = res_miss(1)
        s1_miss(i) = res_miss(2)
        c2_miss(i) = res_miss(3)
        s2_miss(i) = res_miss(4)
        c3_miss(i) = res_miss(5)
        s3_miss(i) = res_miss(6)

        b_4 = transpose(array_t_4(*,i))
        uy_4 = total( b_4 )
        cy_4 = cosp_4 ## (b_4)
        sy_4 = sinp_4 ## (b_4)
```

```
        inh_4 = [ uy_4, cy_4, sy_4]
        res_4 = inv_4#inh_4

        c0_4(i) = res_4(0)
        c1_4(i) = res_4(1)
        s1_4(i) = res_4(2)

        b = transpose(array_t(*,i))
        uy = total( b )
        cy = cosp ## (b)
        sy = sinp ## (b)
        c2y = cos2p ## (b)
        s2y = sin2p ## (b)
        c3y = cos3p ## (b)
        s3y = sin3p ## (b)

        inh = [ uy, cy, sy, c2y, s2y , c3y , s3y ]
        res = inv#inh

        c0(i) = res(0)
        c1(i) = res(1)
        s1(i) = res(2)
        c2(i) = res(3)
        s2(i) = res(4)
        c3(i) = res(5)
        s3(i) = res(6)
endfor

a0 = sqrt( c0^2)
```

```
a1 = sqrt( c1^2 + s1^2)
a2 = sqrt( c2^2 + s2^2)
a3 = sqrt( c3^2 + s3^2)

a0_4 = sqrt( c0_4^2)
a1_4 = sqrt( c1_4^2 + s1_4^2)

a0_miss = sqrt( c0_miss^2)
a1_miss = sqrt( c1_miss^2 + s1_miss^2)
a2_miss = sqrt( c2_miss^2 + s2_miss^2)
a3_miss = sqrt( c3_miss^2 + s3_miss^2)

a0_8=a0
a1_8=a1
a2_8=a2
a3_8=a3

a0_7=a0_miss
a1_7=a1_miss
a2_7=a2_miss
a3_7=a3_miss

;putting data into tree
goto, skip  ; comment to write data to the tree
build_string = 'build_signal(build_with_units($1,"T"),' + $
'*,' + $
'build_dim(build_window(0,' + stremo(ntpts-1) + $
',' + stremo(t(0)) + '),' + $
'build_slope(' + stremo(t(1)-t(0)) + ',,)))'
```

```
mdsput, '\c_m0_p0', build_string, c0
mdsput, '\c_m1_p0', build_string, c1
mdsput, '\c_m2_p0', build_string, c2
mdsput, '\c_m3_p0', build_string, c3
mdsput, '\s_m1_p0', build_string, s1
mdsput, '\s_m2_p0', build_string, s2
mdsput, '\s_m3_p0', build_string, s3

mdsput, '\c_m0_p0_7', build_string, c0_miss
mdsput, '\c_m1_p0_7', build_string, c1_miss
mdsput, '\c_m2_p0_7', build_string, c2_miss
mdsput, '\c_m3_p0_7', build_string, c3_miss
mdsput, '\s_m1_p0_7', build_string, s1_miss
mdsput, '\s_m2_p0_7', build_string, s2_miss
mdsput, '\s_m3_p0_7', build_string, s3_miss

mdsput, '\c_m0_p0_4', build_string, c0_4
mdsput, '\c_m1_p0_4', build_string, c1_4
mdsput, '\s_m1_p0_4', build_string, s1_4

end
```

C.2.5 find_b_wall.pro

```
pro find_b_wall
!p.thick=3
mu_0=4.*!dpi*1e-7
n_theda=360
n_r=11*4
n_ang=24*4
r0=.1
r1=2d-2
curr=1e5
curr=2*!dpi*r0/mu_0

theda=findgen(n_theda)*2*!dpi/n_theda
d_theda=(theda(1)-theda(0))
ds=r0*d_theda
theda_skin=theda+d_theda/2.

;calculating the locations of the filiments and
;    points of interest.
;x,y locations of the electroid, skin, and outer surface
x_elec=r0*cos(theda)
y_elec=r0*sin(theda)
x_skin=(r0-ds)*cos(theda+d_theda/2.)
y_skin=(r0-ds)*sin(theda+d_theda/2.)
x_outer=(r0+ds)*cos(theda+d_theda/2.)
y_outer=(r0+ds)*sin(theda+d_theda/2.)
x_outer=(r0)*cos(theda+d_theda/2.)
y_outer=(r0)*sin(theda+d_theda/2.)
```

```
;x,y location of the electroid, skin, and outer surface
;    relative to the current
x_cur_elec=x_elec-r1
y_cur_elec=y_elec
x_cur_skin=x_skin-r1
y_cur_skin=y_skin
x_cur_outer=x_outer-r1
y_cur_outer=y_outer
theda_cur_elec=atan(y_cur_elec,x_cur_elec)
r_cur_elec=sqrt(y_cur_elec^2+x_cur_elec^2)
theda_cur_skin=atan(y_cur_skin,x_cur_skin)
r_cur_skin=sqrt(y_cur_skin^2+x_cur_skin^2)
theda_cur_outer=atan(y_cur_outer,x_cur_outer)
r_cur_outer=sqrt(y_cur_outer^2+x_cur_outer^2)

;x,y location of the electroid, skin, and outer surface
;    relative to the electrode
x_elec_elec=dblarr(n_theda,n_theda)
y_elec_elec=dblarr(n_theda,n_theda)
for i=0,n_theda-1 do begin
   x_elec_elec(*,i)=x_elec(i)-x_elec
   y_elec_elec(*,i)=y_elec(i)-y_elec
endfor
r_elec_elec=(sqrt(x_elec_elec^2+y_elec_elec^2))>ds

x_elec_skin=dblarr(n_theda,n_theda)
y_elec_skin=dblarr(n_theda,n_theda)
for i=0,n_theda-1 do begin
   x_elec_skin(*,i)=x_skin(i)-x_elec
   y_elec_skin(*,i)=y_skin(i)-y_elec
```

```
endfor
r_elec_skin=(sqrt(x_elec_skin^2+y_elec_skin^2))>ds

x_elec_outer=dblarr(n_theda,n_theda)
y_elec_outer=dblarr(n_theda,n_theda)
for i=0,n_theda-1 do begin
   x_elec_outer(*,i)=x_outer(i)-x_elec
   y_elec_outer(*,i)=y_outer(i)-y_elec
endfor
r_elec_outer=(sqrt(x_elec_outer^2+y_elec_outer^2))>(ds/2.)

;xy location for contour plots
r3=r0*(dindgen(n_r)+1)/(n_r)
a1=dindgen(n_ang)/(n_ang+0.0)*(2*!dpi)

x_int=dblarr(n_ang,n_r)
y_int=dblarr(n_ang,n_r)
for i=0,n_r-1 do begin
  x_int(*,i)=cos(a1)*(r3(i)/r0)* $
            (2*ds+sqrt(r0^2+(cos(a1)*r1)^2-r1^2)-r1*cos(a1))+r1
  y_int(*,i)=sin(a1)*(r3(i)/r0)* $
            (2*ds+sqrt(r0^2+(cos(a1)*r1)^2-r1^2)-r1*cos(a1))
endfor
theda_int=atan(y_int,x_int)
theda_int(where(theda_int lt 0))= $
      theda_int(where(theda_int lt 0))+2*!dpi

x_cur_int=x_int-r1
y_cur_int=y_int
r_cur_int=sqrt((x_cur_int)^2+y_cur_int^2)
```

```
x_elec_int=dblarr(n_theda,n_ang,n_r)
y_elec_int=dblarr(n_theda,n_ang,n_r)
r_elec_int=dblarr(n_theda,n_ang,n_r)
for i=0,n_r-1 do begin
   for j=0,n_ang-1 do begin
      x_elec_int(*,j,i)=x_int(j,i)-x_elec
      y_elec_int(*,j,i)=y_int(j,i)-y_elec
   endfor
endfor
r_elec_int=sqrt(x_elec_int^2+y_elec_int^2)>ds

!p.multi=[0,2,2]
plot,x_elec,y_elec,xra=[-r0-2*ds,r0+2*ds], $
     yra=[-r0-2*ds,r0+2*ds],title='Points of interest'
oplot,x_skin,y_skin,co=2
oplot,x_outer,y_outer,co=4
oplot,x_int,y_int,psym=3,co=3
oplot,[r1,r1],[0,0],psym=4
;caculating the magnetic field from the current
b_x_cur_elec=mu_0*curr/(2*!dpi*r_cur_elec)* $
             (-y_cur_elec/r_cur_elec)
b_y_cur_elec=mu_0*curr/(2*!dpi*r_cur_elec)* $
             ( x_cur_elec/r_cur_elec)
b_r_cur_elec= cos(theda)*b_x_cur_elec+ $
              sin(theda)*b_y_cur_elec
b_t_cur_elec=-sin(theda)*b_x_cur_elec+ $
              cos(theda)*b_y_cur_elec
b_x_cur_skin=mu_0*curr/(2*!dpi*r_cur_skin)* $
             (-y_cur_skin/r_cur_skin)
```

```
b_y_cur_skin=mu_0*curr/(2*!dpi*r_cur_skin)* $
             ( x_cur_skin/r_cur_skin)
b_r_cur_skin= cos(theda_skin)*b_x_cur_skin+ $
              sin(theda_skin)*b_y_cur_skin
b_t_cur_skin=-sin(theda_skin)*b_x_cur_skin+ $
              cos(theda_skin)*b_y_cur_skin
b_x_cur_outer=mu_0*curr/(2*!dpi*r_cur_outer)* $
              (-y_cur_outer/r_cur_outer)
b_y_cur_outer=mu_0*curr/(2*!dpi*r_cur_outer)* $
              ( x_cur_outer/r_cur_outer)
b_r_cur_outer= cos(theda_skin)*b_x_cur_outer+ $
               sin(theda_skin)*b_y_cur_outer
b_t_cur_outer=-sin(theda_skin)*b_x_cur_outer+ $
               cos(theda_skin)*b_y_cur_outer

b_x_cur_int=mu_0*curr/(2*!dpi*r_cur_int)*(-y_cur_int/r_cur_int)
b_y_cur_int=mu_0*curr/(2*!dpi*r_cur_int)*( x_cur_int/r_cur_int)
b_r_cur_int= cos(theda_int)*b_x_cur_int+ $
             sin(theda_int)*b_y_cur_int
b_t_cur_int=-sin(theda_int)*b_x_cur_int+ $
             cos(theda_int)*b_y_cur_int

plot,theda,b_t_cur_elec,yra=([-0,1.1]*max(b_t_cur_elec)), $
     title='B!d!4h!3!n from the current', $
     xtitle='!4h!3 (rad)', ytitle='B(T)'
oplot,theda_skin,b_t_cur_skin,co=2
oplot,theda_skin,b_t_cur_outer,co=4
oplot,theda_int(*,n_r-1),b_t_cur_int(*,n_r-1),co=3

plot,theda,b_r_cur_elec,title='B!dr!n from the current', $
```

```
     xtitle='!4h!3 (rad)', ytitle='B(T)'
oplot,theda_skin,b_r_cur_skin,co=2
oplot,theda_skin,b_r_cur_outer,co=4
oplot,theda_int(*,n_r-1),b_r_cur_int(*,n_r-1),co=3

levels=(findgen(10)+1)*.9
c_colors=[12,4,11,5,10,3,9,8,6,2]

;initial guess of the current profile
j_elec=dblarr(n_theda)-curr/(2*!dpi*r0)
;Calculating the magnetic field due to the electrode
b_y_elec_elec= (ds*mu_0/(2*!dpi))* $
            j_elec#(x_elec_elec/(r_elec_elec^2))
b_x_elec_elec=-(ds*mu_0/(2*!dpi))* $
            j_elec#(y_elec_elec/(r_elec_elec^2))
b_r_elec_elec= cos(theda)*b_x_elec_elec+ $
               sin(theda)*b_y_elec_elec
b_t_elec_elec=-sin(theda)*b_x_elec_elec+ $
               cos(theda)*b_y_elec_elec

b_y_elec_skin= (ds*mu_0/(2*!dpi))* $
       j_elec#(x_elec_skin/(r_elec_skin^2))
b_x_elec_skin=-(ds*mu_0/(2*!dpi))* $
       j_elec#(y_elec_skin/(r_elec_skin^2))
b_r_elec_skin= cos(theda_skin)*b_x_elec_skin+ $
               sin(theda_skin)*b_y_elec_skin
b_t_elec_skin=-sin(theda_skin)*b_x_elec_skin+ $
               cos(theda_skin)*b_y_elec_skin

b_y_elec_outer= (ds*mu_0/(2*!dpi))* $
```

```
       j_elec#(x_elec_outer/(r_elec_outer^2))
b_x_elec_outer=-(ds*mu_0/(2*!dpi))* $
       j_elec#(y_elec_outer/(r_elec_outer^2))
b_r_elec_outer= cos(theda_skin)*b_x_elec_outer+ $
                sin(theda_skin)*b_y_elec_outer
b_t_elec_outer=-sin(theda_skin)*b_x_elec_outer+ $
                cos(theda_skin)*b_y_elec_outer

b_y_elec_int=dblarr(n_ang,n_r)
b_x_elec_int=dblarr(n_ang,n_r)
for i=0,n_r-1 do begin
   b_y_elec_int(*,i)= (ds*mu_0/(2*!dpi))* $
          j_elec#(x_elec_int(*,*,i)/(r_elec_int(*,*,i)^2))
   b_x_elec_int(*,i)=-(ds*mu_0/(2*!dpi))* $
          j_elec#(y_elec_int(*,*,i)/(r_elec_int(*,*,i)^2))
endfor
b_r_elec_int= cos(theda_int)*b_x_elec_int+ $
              sin(theda_int)*b_y_elec_int
b_t_elec_int=-sin(theda_int)*b_x_elec_int+ $
              cos(theda_int)*b_y_elec_int

;finding the total field

b_t_total_skin=b_t_elec_skin+b_t_cur_skin
b_r_total_skin=b_r_elec_skin+b_r_cur_skin
b_t_total_elec=b_t_elec_elec+b_t_cur_elec
b_r_total_elec=b_r_elec_elec+b_r_cur_elec
b_t_total_outer=b_t_elec_outer+b_t_cur_outer
b_r_total_outer=b_r_elec_outer+b_r_cur_outer
```

```
s_b_t_total_skin=b_t_total_skin

b_t_total_int=b_t_elec_int+b_t_cur_int
b_r_total_int=b_r_elec_int+b_r_cur_int

levels=(findgen(10)+1)*.9
c_colors=[12,4,11,5,10,3,9,8,6,2]
contour,sqrt(b_t_cur_int^2+b_r_cur_int^2),x_int,y_int, $
        /irregular,c_colors=c_colors,levels=levels, $
        title='B!dtotal!n from the current'
oplot,x_elec,y_elec

dbr_dt=fltarr(n_theda)
dbr_dt(1:n_theda-1)=b_r_total_skin(1:n_theda-1)- $
                    b_r_total_skin(0:n_theda-2)
dbr_dt(0)=b_r_total_skin(0)-b_r_total_skin(n_theda-1)
dbr_dt=-dbr_dt/(d_theda*r0*mu_0)

count=0
!p.multi=0
plot,theda_skin,b_r_total_skin,title='B!dr!n total at the skin'
while (max(abs(dbr_dt*(2*ds)) gt (min(abs(j_elec))*1e-4))) and $
        (count le n_theda) do begin
   j_elec=j_elec-dbr_dt*(2*ds)

   b_y_elec_skin= (ds*mu_0/(2*!dpi))* $
          j_elec#(x_elec_skin/(r_elec_skin^2))
   b_x_elec_skin=-(ds*mu_0/(2*!dpi))* $
          j_elec#(y_elec_skin/(r_elec_skin^2))
   b_r_elec_skin= cos(theda_skin)*b_x_elec_skin+ $
```

```
                    sin(theda_skin)*b_y_elec_skin

   b_r_total_skin=b_r_elec_skin+b_r_cur_skin

   dbr_dt=fltarr(n_theda)
   dbr_dt(1:n_theda-1)=b_r_total_skin(1:n_theda-1)- $
                       b_r_total_skin(0:n_theda-2)
   dbr_dt(0)=b_r_total_skin(0)-b_r_total_skin(n_theda-1)
   dbr_dt=-dbr_dt/(d_theda*r0*mu_0)
   oplot,theda_skin,b_r_total_skin,co=2+(count/(.1*n_theda))
   count=count+1
endwhile
;Calculating the magnetic field due to the electrode
b_y_elec_elec= (ds*mu_0/(2*!dpi))* $
       j_elec#(x_elec_elec/(r_elec_elec^2))
b_x_elec_elec=-(ds*mu_0/(2*!dpi))* $
       j_elec#(y_elec_elec/(r_elec_elec^2))
b_r_elec_elec= cos(theda)*b_x_elec_elec+ $
               sin(theda)*b_y_elec_elec
b_t_elec_elec=-sin(theda)*b_x_elec_elec+ $
               cos(theda)*b_y_elec_elec

b_y_elec_skin= (ds*mu_0/(2*!dpi))* $
       j_elec#(x_elec_skin/(r_elec_skin^2))
b_x_elec_skin=-(ds*mu_0/(2*!dpi))* $
       j_elec#(y_elec_skin/(r_elec_skin^2))
b_r_elec_skin= cos(theda_skin)*b_x_elec_skin+ $
               sin(theda_skin)*b_y_elec_skin
b_t_elec_skin=-sin(theda_skin)*b_x_elec_skin+ $
               cos(theda_skin)*b_y_elec_skin
```

```
b_y_elec_outer= (ds*mu_0/(2*!dpi))* $
       j_elec#(x_elec_outer/(r_elec_outer^2))
b_x_elec_outer=-(ds*mu_0/(2*!dpi))* $
       j_elec#(y_elec_outer/(r_elec_outer^2))
b_r_elec_outer= cos(theda_skin)*b_x_elec_outer+ $
                sin(theda_skin)*b_y_elec_outer
b_t_elec_outer=-sin(theda_skin)*b_x_elec_outer+ $
                cos(theda_skin)*b_y_elec_outer

b_y_elec_int=dblarr(n_ang,n_r)
b_x_elec_int=dblarr(n_ang,n_r)
for i=0,n_r-1 do begin
   b_y_elec_int(*,i)= (ds*mu_0/(2*!dpi))* $
         j_elec#(x_elec_int(*,*,i)/(r_elec_int(*,*,i)^2))
   b_x_elec_int(*,i)=-(ds*mu_0/(2*!dpi))* $
         j_elec#(y_elec_int(*,*,i)/(r_elec_int(*,*,i)^2))
endfor
b_r_elec_int= cos(theda_int)*b_x_elec_int+ $
              sin(theda_int)*b_y_elec_int
b_t_elec_int=-sin(theda_int)*b_x_elec_int+ $
              cos(theda_int)*b_y_elec_int

!p.multi=[0,2,2]
levels=(findgen(10)+1)*.9
c_colors=[12,4,11,5,10,3,9,8,6,2]
contour,sqrt(b_t_cur_int^2+b_r_cur_int^2),x_int,y_int, $
        /irregular,c_colors=c_colors,levels=levels, $
        title='B!dtotal!n from the current'
oplot,x_elec,y_elec
```

```
levels=levels*.1
contour,sqrt(b_t_elec_int^2+b_r_elec_int^2),x_int,y_int, $
        /irregular,c_colors=c_colors,levels=levels, $
        title='B!dtotal!n from the wall (levels*.1)'
oplot,x_elec,y_elec

;finding the total field
b_x_total_int=b_x_elec_int+b_x_cur_int
b_y_total_int=b_y_elec_int+b_y_cur_int
b_t_total_skin=b_t_elec_skin+b_t_cur_skin
b_r_total_skin=b_r_elec_skin+b_r_cur_skin
b_t_total_elec=b_t_elec_elec+b_t_cur_elec
b_r_total_elec=b_r_elec_elec+b_r_cur_elec
b_t_total_outer=b_t_elec_outer+b_t_cur_outer
b_r_total_outer=b_r_elec_outer+b_r_cur_outer

b_t_total_int=b_t_elec_int+b_t_cur_int
b_r_total_int=b_r_elec_int+b_r_cur_int

levels=(findgen(10)+1)*.9
c_colors=[12,4,11,5,10,3,9,8,6,2]
contour,sqrt(b_t_total_int^2+b_r_total_int^2), $
        x_int,y_int,/irregular, $
        c_colors=c_colors,levels=levels, $
        title='B!dtotal!n from both'
oplot,x_elec,y_elec
levels=levels*.1
contour,b_y_elec_int,x_int,y_int,/irregular, $
        c_colors=c_colors,levels=levels, $
```

```
        title='B!dy!n from the wall (levels*.1)'
oplot,x_elec,y_elec

!p.multi=0
plot,theda_skin,b_t_total_skin,XRA=[0,2*!pi],xstyle=1, $
        title='B!d!4h!3!n from both',xtitle='!4h!3 (rad)', $
        ytitle='B(T)'
oplot,theda_skin,s_b_t_total_skin,co=2
hline,mu_0*curr/(2*!pi*(r0-ds)),co=3
;read,bogus
oplot,theda_skin,2*b_t_cur_skin-mu_0*curr/(2*!pi*(r0-ds)), $
      co=4,linestyle=2

ps=1
if ps eq 1 then begin
  SET_PLOT, 'PS'
  DEVICE, FILENAME='find_b_wall.ps', /color , $
          /portrait,ysize=10,yoffset=1.2
  !p.thick=3
  !x.thick=3
  !y.thick=3
  !p.charthick=2
  !p.charsize=1
endif

plot,theda_skin,b_t_total_skin,xra=[0,2*!pi],xstyle=1, $
     title='Constant Displacement',xtitle='!4h!3 (rad)', $
     ytitle='B(T)'
oplot,theda_skin,s_b_t_total_skin,co=2
hline,mu_0*curr/(2*!pi*(r0-ds)),co=3
```

```
oplot,theda_skin,2*b_t_cur_skin-mu_0*curr/(2*!pi*(r0-ds)), $
      co=4,linestyle=2

xyouts,.7,.85,'B!d!4h!3!u0!n+ B!d!4h!3!u1!n+B!d!4h!3!u2!n', $
       /normal
xyouts,.7,.8,'B!d!4h!3!u0!n',/normal,co=3
xyouts,.7,.75,'B!d!4h!3!u0!n+ B!d!4h!3!u1!n',/normal,co=2
xyouts,.7,.7,'B!d!4h!3!u0!n+2B!d!4h!3!u1!n',/normal,co=4

if ps eq 1 then begin
  device,/close
  SET_PLOT, 'win'
  !p.thick=1
  !x.thick=1
  !y.thick=1
  !p.charthick=1
  !p.charsize=1.5
  !p.multi=0
endif
!p.multi=0

end
```

C.3 Density Codes

The density is calculated and written to the tree with *Dense_fin.pro*. The code assumes the tree is already open. The density is calculated with the method described in Sec. 5.6.3. The fringe jumps are found with a series of tests. This method removes any bias which may occur when an operator determines the fringe jumps. The phase may be lost when the

radius of the lissajou becomes zero. The code can normally find this point and correct it when the radius goes through zero once during a pulse. The radius of the lissajou should be checked when the density has a $2\pi 5.61 \times 10^{20}$ m^{-2} jump for a period of time. The code may miss two fringe jumps when the radius goes through zero twice.

C.3.1 Dense_fin.pro

```
pro dense_fin
;
;getting shot
;
shot = mdsvalue('$shot')
;
; Conversion using 632.8nm for laser wavelength, eq 4.2 in Hermann Thesis
;
convert=5.61
;
;chord length of each view
;
l=dblarr(4)
l(0)=0.0938
l(1)=0.124
l(2)=0.145
l(3)=19.261
;
;Setting initial noise end to I_P = 50000
;
Ip=mdsvalue('\I_P')
Tip=mdsvalue('dim_of(\I_P)')
ip=abs(dsmooth(ip,5,7))
```

```
Kip=min(where(ip gt 2e4))
If (kip eq -1) then kip=0
Noise_end = tip(kip)*1e6

Kip=max(where(ip gt 2e4))
If (kip eq -1) then kip=max(where(Tip gt 200e-6))
Plasma_end = tip(kip)*1e6 + 200
;
; Digital filter parameters
;
Flow=0
Fmid=.2
Fhigh=1
a=50
N_terms=13
Coeff_low = DIGITAL_FILTER(Flow, Fmid, A, N_terms)
Coeff_high= DIGITAL_FILTER(Fmid, Fhigh, A, N_terms)
;
; Magnitude of phi change to account for fringe jumps
;
jump=!pi
;
;Do loop for 2 chords
;
for chord=1,2 do begin
   ;
   ;Getting time base, location, sin and cos values
   ;
   if (chord eq 1) then begin
```

```
  cos1=mdsvalue('\digitizers::cos1')
  sin1=mdsvalue('\digitizers::sin1')
  to=mdsvalue('dim_of(\digitizers::cos1)')
  loc=mdsvalue('\NE_1:HOLE_LOC')
  baseline=mdsvalue('\ne_1:baseline')
  z=mdsvalue('\ne_1:z_loc')
endif else begin
  cos1=mdsvalue('\digitizers::cos2')
  sin1=mdsvalue('\digitizers::sin2')
  to=mdsvalue('dim_of(\digitizers::cos2)')
  loc=mdsvalue('\NE_2:HOLE_LOC')
  baseline=mdsvalue('\ne_2:baseline')
  z=mdsvalue('\ne_2:z_loc')
endelse
;
; Adjusting time base
;
to=to*1e6
;
;Get the array size
;
ndata=n_elements(cos1)
;
;Offset method
;
if ( shot ne baseline) then begin
  mdsopen,'zapmain',stremo(baseline)
  if (chord eq 1) then begin
    xpo=mdsvalue('\ne_1:sin_offset')
    ypo=mdsvalue('\ne_1:cos_offset')
```

```
    del_xpo=mdsvalue('\ne_1:sin_offset:error')
    del_ypo=mdsvalue('\ne_1:cos_offset:error')
  endif else begin
    xpo=mdsvalue('\ne_2:sin_offset')
    ypo=mdsvalue('\ne_2:cos_offset')
    del_xpo=mdsvalue('\ne_2:sin_offset:error')
    del_ypo=mdsvalue('\ne_2:cos_offset:error')
  endelse
  mdsopen,'zapmain',stremo(shot)
endif else begin
  xpo=total(sin1)/ndata
  ypo=total(cos1)/ndata
  del_xpo=sqrt(total((sin1-xpo)^2)/(ndata-1))
  del_ypo=sqrt(total((cos1-ypo)^2)/(ndata-1))
  sin1=sin1+1.
  cos1=cos1+1.
endelse
;
; Adjustment for center offset
;
xo=sin1-xpo
yo=cos1-ypo
;
; Rotating the coordinate system
;
theda_0=atan(total(xo(0:10))/11,total(yo(0:10))/11)
xo_t=xo
yo_t=yo
xo=xo_t*cos(theda_0)-yo_t*sin(theda_0)
yo=yo_t*cos(theda_0)+xo_t*sin(theda_0)
```

```
;
; Calculate the phase shift
;
y=atan(xo,yo)
;
; Calculate the phase shift with applying the digital filter
;
x_low=CONVOL(xo, Coeff_low)
y_low=CONVOL(yo, Coeff_low)
x_high=CONVOL(xo, Coeff_high)
y_high=CONVOL(yo, Coeff_high)

y_sm=atan(x_low,y_low)
to_sm=to
ndata_sm=n_elements(to_sm)
;
;Correction for VAX atan(0,0)
n_xy_zero=where(((x_low eq 0) and (y_low eq 0)) and (to le 0))
y_sm(n_xy_zero)=y_sm(max(n_xy_zero)+2)
n_xy_zero=where(((x_low eq 0) and (y_low eq 0)) and (to ge 0))
y_sm(n_xy_zero)=y_sm(min(n_xy_zero)-2)
;
;getting the amplitude of the noise
;
x_high1=dsmooth(abs(x_high))+sqrt((del_xpo*cos(theda_0))^2+ $
                  (del_ypo*sin(theda_0))^2)
y_high1=dsmooth(abs(y_high))+sqrt((del_ypo*cos(theda_0))^2+ $
                  (del_xpo*sin(theda_0))^2)
;
;finding errors in psi
```

```
;
r_2=(x_low^2+y_low^2)>(del_xpo^2+del_ypo^2)
del_y=sqrt(((yo*x_high1)^2+(xo*y_high1)^2)/((r_2)^2))
del_y=del_y<(2*!dpi)
;
;Finding earliest vibration fringe jump
;
ave1=total(y_sm(1:50))/50.
n_time_m1=min(where(to_sm ge -1e-6))>50
ave2=total(y_sm(n_time_m1-50:n_time_m1-1))/50.
ave_slope=(ave2-ave1)/(to_sm(n_time_m1-25)-to_sm(25))
t_end_slope=to(25)+(2.5-abs(ave1))/abs(ave_slope)
;
; Finding region for vibration fits
;
k=min(where(to ge noise_end))
pk=min(where(to ge plasma_end<t_end_slope))
;
; Accounting for fringe jumps after the plasma pulse
;
a=dblarr(ndata-pk)
b=dblarr(ndata-pk)
d=dblarr(ndata-pk)
a=(y(pk:ndata-2)-y((pk+1):ndata-1)) gt jump
b=(y(pk:ndata-2)-y((pk+1):ndata-1)) lt -jump
d=1.0*a-1.0*b
sum=dblarr(ndata-pk)
for I=1,ndata-1-pk do $
   sum(I)=sum(I-1)+d(I-1)
y(pk+1:ndata-1)=y(pk+1:ndata-1)+2.0*!pi*sum(1:ndata-pk-1)
```

```
a=dblarr(ndata_sm-pk)
b=dblarr(ndata_sm-pk)
d=dblarr(ndata_sm-pk)
a=(y_sm(pk:ndata_sm-2)-y_sm(pk+1:ndata_sm-1)) gt jump
b=(y_sm(pk:ndata_sm-2)-y_sm(pk+1:ndata_sm-1)) lt -jump
d=1.0*a-1.0*b
sum=dblarr(ndata_sm-pk)
for I=1,ndata_sm-1-pk do $
   sum(I)=sum(I-1) + d(I-1)
y_sm(pk+1:ndata_sm-1)=y_sm(pk+1:ndata_sm-1)+2.0*!pi*sum(1:ndata_sm-pk-1)
;
pk=min(where(to ge plasma_end))
;
;finding time for baseline
;
nset=[where(to_sm le 2e-6),where(to_sm ge plasma_end)]
;
; Finding baseline and error
;
diag=INDGEN(5)*(5 + 1)
pder=[[replicate(1., n_elements(nset))],[to(nset)], $
      [to(nset)^2],[to(nset)^3],[to(nset)^4]]
weight=1/(del_y(nset)^2)
alpha = transpose(pder) # (weight# (fltarr(5)+1.)*pder)
alpha_inv=invert(alpha)
drift_coeff=invert(transpose(pder) # pder) # $
                  (transpose(pder) # y_sm(nset))
sigma = sqrt( alpha_inv[diag] )
del_phi_vib=sigma(0)+to*sigma(1)+(to^2)*sigma(2)+ $
                 (to^3)*sigma(3)+(to^4)*sigma(4)
```

```
del_phi_p=sqrt(del_y^2+del_phi_vib^2)
del_phi_p(0:20)=0
del_phi_p(ndata-21:ndata-1)=0
drift=drift_coeff(0)+to*drift_coeff(1)+(to^2)*drift_coeff(2)+ $
                (to^3)*drift_coeff(3)+(to^4)*drift_coeff(4)
y=y-drift
y_sm=y_sm-drift
;
; Modified Milroy code using Uri's idea for accounting for
;          fringe jumps (Jul 03)
;
nstart=min(where(to ge 0))
a1=fltarr(ndata)
b1=fltarr(ndata)
a1(1:ndata-1)=(y(0:ndata-2)-y(1:ndata-1)) ge jump
b1(1:ndata-1)=(y(0:ndata-2)-y(1:ndata-1)) le -jump
d1=1.0*a1-1.0*b1
d1(0:nstart+100)=0
sumf=fltarr(ndata)
for i=1,ndata-1 do $
   sumf(i)=sumf(i-1)+d1(i)
sumb=fltarr(ndata)
for i=ndata-2,0,-1 do $
   sumb(i)=sumb(i+1)+d1(i+1)
phi_f=y+2*!pi*sumf
phi_b=y-2*!pi*sumb
phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
a_new=phi_f - phi_b_new
cross= 1.*(phi_b(ndata-100) gt phi_f(ndata-100))- $
```

```
          1.*(phi_b(ndata-100) lt phi_f(ndata-100))
   count=0
   if cross gt 0 then begin
     phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
 a_new=phi_f - phi_b_new
 while (cross ne 0) and (count le 15) do begin
fringe=max(where(a_new(0:ndata-200)*cross gt -1*!pi))
new_sum=fltarr(ndata)
new_sum(fringe:ndata-1)=2*!pi*cross
phi_f=phi_f+new_sum
phi_b=phi_b-2*!pi*cross +new_sum
phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
a_new=phi_f - phi_b_new
cross= 1.*(phi_b(ndata-100) gt phi_f(ndata-100))- $
                   1.*(phi_b(ndata-100) lt phi_f(ndata-100))
        count=count+1
 endwhile
   endif else begin
     phi_b_new= -phi_b
     a_new=phi_f - phi_b_new
 while (cross ne 0) and (count le 15) do begin
fringe=max(where(a_new(0:ndata-200)*cross gt 1.*!pi))+1
new_sum=fltarr(ndata)
new_sum(fringe:ndata-1)=2*!pi*cross
phi_f=phi_f+new_sum
phi_b=phi_b-2*!pi*cross +new_sum
phi_b_new= -phi_b
     a_new=phi_f - phi_b_new
cross=- 1.*((phi_f(0)-phi_b(0)) gt .001)
count=count+1
```

```
endwhile
  endelse
  y=-phi_b_new
  ;
  ;Modified Milroy code using Uri's idea for accounting for
  ;         fringe jumps w/filtered data (Jul 03)
  ;
  a1=fltarr(ndata_sm)
  b1=fltarr(ndata_sm)
  a1(1:ndata_sm-1)=(y_sm(0:ndata_sm-2)-y_sm(1:ndata_sm-1)) gt jump
  b1(1:ndata_sm-1)=(y_sm(0:ndata_sm-2)-y_sm(1:ndata_sm-1)) lt -jump
  d1=1.0*a1-1.0*b1
  d1(0:nstart+100)=0
  sumf=fltarr(ndata_sm)
  for i=1,ndata_sm-1 do $
     sumf(i)=sumf(i-1)+d1(i)
  sumb=fltarr(ndata_sm)
  for i=ndata_sm-2,0,-1 do $
     sumb(i)=sumb(i+1)+d1(i+1)
  phi_f=y_sm+2*!pi*sumf
  phi_b=y_sm-2*!pi*sumb

  phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
  a_new=phi_f - phi_b_new
  cross= 1.*(phi_b(ndata_sm-100) gt phi_f(ndata_sm-100))- $
         1.*(phi_b(ndata_sm-100) lt phi_f(ndata_sm-100))
  count=0
  if cross gt 0 then begin
    phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
a_new=phi_f - phi_b_new
```

```
 while (cross ne 0) and (count le 10) do begin
fringe=max(where(a_new(0:ndata_sm-200)*cross gt -1*!pi))
new_sum=fltarr(ndata_sm)
new_sum(fringe:ndata_sm-1)=2*!pi*cross
phi_f=phi_f+new_sum
phi_b=phi_b-2*!pi*cross +new_sum
phi_b_new= -phi_b+ phi_f(0)+phi_b(0)
a_new=phi_f - phi_b_new
cross= 1.*(phi_b(ndata_sm-100) gt phi_f(ndata_sm-100))- $
                   1.*(phi_b(ndata_sm-100) lt phi_f(ndata_sm-100))
        count=count+1
 endwhile
   endif else begin
     phi_b_new= -phi_b
     a_new=phi_f - phi_b_new
 while (cross ne 0) and (count le 10) do begin
fringe=max(where(a_new(0:ndata_sm-200)*cross gt 1.*!pi))+1
new_sum=fltarr(ndata_sm)
new_sum(fringe:ndata_sm-1)=2*!pi*cross
phi_f=phi_f+new_sum
phi_b=phi_b-2*!pi*cross +new_sum
    phi_b_new= -phi_b
     a_new=phi_f - phi_b_new
cross=- 1.*((phi_f(0)-phi_b(0)) gt .001)
count=count+1
 endwhile
   endelse
   y_sm=-phi_b_new
   ;
   ; forcing all densities to be positive
```

```
;
y(where(y le -.5)>0)=y(where(y le -.5)>0)+2*!pi
y_sm(where(y_sm le -.5)>0)=y_sm(where(y_sm le -.5)>0)+2*!pi
;
;forcing last point to be a fringe jump
;
y(ndata-1)=2*!pi
y_sm(ndata_sm-1)=2*!pi
;
; Converting to density
;
y=y*convert
y_sm=y_sm*convert
del_phi_p=del_phi_p*convert
;stop
;
; Putting data unto the tree
;
build_string = 'build_signal(build_with_units($1*1e20,"m^(-2)"),' + $
     '*,(build_with_units($2*1e-6,"s")))'
build_string_sm = 'build_signal(build_with_units($1*1e20/' + $
     stremo(l(loc-1))+',"m^(-2)"),' + $
     '*,(build_with_units($2*1e-6,"s")))'
build_string_with_err = $
     'build_signal(build_with_errors(build_with_units' + $
     '($1*1e20,"m^(-2)"), $3)' + $
'*, (build_with_units($2*1e-6,"s")))'
mdsopen,'zapmain',stremo(shot)
if (chord eq 1) then begin
  mdsput, '\ne_1', build_string, y, to
```

```
      mdsput, '\ne_1:ne_1_sm', build_string, y_sm, to
      mdsput, '\ne_1:ne_1_ave', build_string_sm, y, to
      mdsput, '\ne_1:ne_1_sm_ave', build_string_sm, y_sm, to
      mdsput, '\ne_1:fmid', '$', Fmid
      mdsput, '\ne_1:jump', '$', jump
      mdsput, '\ne_1:sin_offset', '$', xpo
      mdsput, '\ne_1:sin_offset:error', '$', del_xpo
      mdsput, '\ne_1:cos_offset', '$', ypo
      mdsput, '\ne_1:cos_offset:error', '$', del_ypo
      mdsput, '\ne_1:error', build_string, del_phi_p, to
    endif else begin
      mdsput, '\ne_2', build_string, y, to
      mdsput, '\ne_2:ne_2_sm', build_string, y_sm, to
      mdsput, '\ne_2:ne_2_ave', build_string_sm, y, to
      mdsput, '\ne_2:ne_2_sm_ave', build_string_sm, y_sm, to
      mdsput, '\ne_2:fmid', '$', Fmid
      mdsput, '\ne_2:jump', '$', jump
      mdsput, '\ne_2:sin_offset', '$', xpo
      mdsput, '\ne_2:sin_offset:error', '$', del_xpo
      mdsput, '\ne_2:cos_offset', '$', ypo
      mdsput, '\ne_2:cos_offset:error', '$', del_ypo
      mdsput, '\ne_2:error', build_string, del_phi_p, to
    endelse
endfor
end
;
; Past comments and code
;

;--Finding the center of the DC offsets (No longer used 7 Feb 99)
```

```
;  [;xpo=total(sin1(0:ndata-1))/ndata]
;  [;ypo=total(cos1(0:ndata-1))/ndata]
;  [;xp=sin1-xpo]
;  [;yp=cos1-ypo]
;  [;xpp=(total(xp^3+xp*yp^2)*total(yp^2) - $]
;  [;    (total(yp^3+yp*xp^2))*total(xp*yp))$]
;  [;     /(2*(total(xp^2)*total(yp^2)-(total(xp*yp))^2))]
;  [;ypp=(total(yp^3+yp*xp^2)*total(xp^2) - $]
;  [;    (total(xp^3+xp*yp^2))*total(xp*yp))$]
;  [;     /(2*(total(xp^2)*total(yp^2)-(total(xp*yp))^2))]
;
;--rotating the coordinate system, added 4jun02
;
;--added to offset method 4jun02:
;  [;sin1=fltarr(ndata)+1.;cos1=fltarr(ndata)+1. ]
```

C.4 Spectroscopy Codes

The plasma velocity has been measured with three spectrometers. The data analysis codes for the IDS and ICCD spectrometers are included in this section.

C.4.1 Calibration codes

The IDS calibration factors are found with *ids_cal4.pro*. The method used to calculate the calibration factors was discussed in Sec. 5.7.3. The code calculates the calibration factors for a pulse and prints them to the screen. The calibration factors are recorded and used in ids_temp_fin3.pro.

The calibration of the ICCD was discussed in Sec. 5.7.4. A large number of calibration sets are needed for the ICCD. These sets are with hundreds of spectra. An example code, *iccd_18sep01_cal_600.pro*, is used to find the ICCD calibration factors. The code plots the calibration factors from all the images for many of the steps used to find the calibration factors into a file. The calibration factor are check by reviewing the plots in the file. When the calibration factors calculated by the code are incorrect, such as the number of rows in a bin, modifications are made to the code or the calibration factors are calculated by hand. Once all of the calibration factors have been verified, *save_cal_18sep01_param.pro* averages the calibration factors and saves them to a file. The code is verified by calculating the calibration factors by hand for the wavelength regions where velocity measurements are made.

ids_cal4.pro

```
pro ids_cal4,shot=shot,par=par,sig=sig

default,shot,315012
ss,shot

ids1=data('\digitizers::zap_3412_a:input_1',xaxis=tim)

nel=n_elements(ids1)
nch=16
sm=7
w=120

d_t=(tim(2)-tim(0))/2
n_os=1/(d_t*w)

ids1_sm=smooth(ids1,sm)-.1
totn_os=nel/n_os
w_os=fltarr(totn_os/4)

for i=0,totn_os/4-1 do begin
  mint=0+n_os*i
  maxt=n_os-1+n_os*i
  lds_max=where(ids1_sm(mint:maxt) ge 1.001*max(ids1_sm(mint:maxt)))
  lds_min=where(ids1_sm(mint:maxt) le .999*min(ids1_sm(mint:maxt)))
  wc=.5*1/abs(tim(lds_min(n_elements(lds_min)/2)) -  $
              tim(lds_max(n_elements(lds_max)/2)))
  w_os(i)=wc
```

```
endfor

ave=moment(w_os)
n_os=1/(d_t*w)
totn_os=nel/n_os

n_skip=n_os*findgen(totn_os)
tot=fltarr(nel/totn_os)
ids_peak=fltarr(nch,totn_os)
ids=fltarr(nch,nel)
ids_sm=fltarr(nch,nel)
ids_fit=fltarr(nch,nel-1)
n_mintot=fltarr(nch)

ch_nam=['\digitizers::zap_3412_a:input_1', $
        '\digitizers::zap_3412_a:input_2', $
        '\digitizers::zap_3412_a:input_3', $
        '\digitizers::zap_3412_a:input_4', $
        '\digitizers::zap_3412_b:input_1', $
        '\digitizers::zap_3412_b:input_2', $
        '\digitizers::zap_3412_b:input_3', $
        '\digitizers::zap_3412_b:input_4', $
        '\digitizers::zap_3412_c:input_1', $
        '\digitizers::zap_3412_c:input_2', $
        '\digitizers::zap_3412_c:input_3', $
        '\digitizers::zap_3412_c:input_4', $
        '\digitizers::zap_3412_d:input_1', $
        '\digitizers::zap_3412_d:input_2', $
        '\digitizers::zap_3412_d:input_3', $
        '\digitizers::zap_3412_d:input_4']
```

```
par=fltarr(nch,3)
sig=fltarr(nch,3)

for i = 0, nch-1 do begin
   ids(i,0:nel-1)=data(ch_nam(i))
   ids_sm(i,0:nel-1)=smooth(ids(i,0:nel-1),sm)
   for j=0,n_os-1 do begin
      tot_tmp=transpose(ids_sm(i,n_skip+j))
      tot(j)=min(tot_tmp)
   endfor
   n_mintot(i)=min(where(tot eq min(tot)))
   ids_peak(i,0:totn_os-1)=ids_sm(i,n_skip+n_mintot(i))
   y=transpose(ids_peak(i,*))
   x=tim(n_skip+n_mintot(i))
   plot,x,y
   yfit=gaussfit(x,y,b,nterms=4)  ;Intitial guess using guassfit
   oplot,x,yfit,color=2
   a=b(0:2)
   w=fltarr(n_elements(x))+1./(0.002^2)
   yfit(*)=curvefit(x,y,w,a,sigma,function_name='fgauss3')
   oplot,x,yfit,color=3
   par(i,*)=a
   sig(i,*)=sigma
endfor

ip=fltarr(totn_os)

for i = 0, nch-1 do begin
   ip=transpose(ids_peak(i,0:totn_os-1))
   t_skip=tim(n_skip+n_mintot(i))
```

```
   idspline=spline(t_skip,ip,tim)
   ids_fit(i,*)=idspline
endfor

peak_tm=fltarr(nch)
peak_min=fltarr(nch)
fwhm=fltarr(nch)
disp=fltarr(nch-1)

for i=0,nch-1 do begin
    n_max=where(transpose(ids_fit(i,*)) eq min(ids_fit(i,*)))
    n_m=n_max(n_elements(n_max)/2)
    peak_tm(I)=tim(n_m)
    start=max(where( transpose(ids_fit(I,*)) gt -0.02))
    start_cen=min(where( transpose(ids_fit(I,*)) gt -0.02))
    mom=int_tabulated(tim(start_cen:start), $
          tim(start_cen:start)* $
          transpose(ids_fit(i,start_cen:start)))
    aera=int_tabulated(tim(start_cen:start), $
          transpose(ids_fit(i,start_cen:start)))
    cen=mom/aera
    peak_tm(i)=cen
    fwh=where(transpose(ids_fit(I,*)) lt .5*ids_fit(I,n_m))
    start=max(fwh)
    start_cen=min(fwh)
    mom=int_tabulated(tim(start_cen:start), $
         tim(start_cen:start)* $
         transpose(ids_fit(i,start_cen:start)))
    aera=int_tabulated(tim(start_cen:start), $
          transpose(ids_fit(i,start_cen:start)))
```

```
    cen=mom/aera
    peak_tm(i)=cen
    fwh=where(transpose(ids_fit(I,*)) lt .5*ids_fit(I,n_m))
    fwhm(I)=tim(max(fwh))-tim(min(fwh))
    peak_min(I)=(ids_fit(i,min(where(tim ge peak_tm(i)))))
endfor

disp=peak_tm(1:nch-1)-peak_tm(0:nch-2)

print,'Shot=',shot
print,'time=',peak_tm
print,'time_fit=',(par(*,1))
print,'time_sig=',(sig(*,1))
print,'peak=',peak_min
print,'peak_fit=',(par(*,0))
print,'peak_sig=',(sig(*,0))
print,'FWHM=',fwhm
print,'FWHM_fit=',(par(*,2))
print,'FWHM_sig=',(sig(*,2))
print,'Disp=',disp
print,'Disp_fit=',(par(1:nch-1,1)-par(0:nch-2,1))
print,'Disp_sig=',((sig(1:nch-1,1)*sig(1:nch-1,1)+ $
                  par(0:nch-2,1)*par(0:nch-2,1))^.5)
print,'Rel Int (ch8)=',peak_min/peak_min(7)
print,'Rel_Int_fit=',(par(*,0)/par(7,0))
print,'Rel_Int_sig=',((sig(*,0)/par(7,0))^2+ $
                 (sig(7,0)*par(*,1)/(par(7,1)^2))^2)^.5
end
```

iccd_18sep01_cal_600.pro

```
;pro iccd_18sep01_cal_600
default,ha_int,0

if ha_int ne 1 then begin
  n_shot=300 ;total calibration shots
  ;n_shot=6
  n_el=512 ;n pixels in the picture
  n_chord=20 ;number of chords
  image_all=fltarr(n_el,n_el,n_shot)
  wave_all=fltarr(n_el,n_shot)
endif

r_level=.1

; initializing arrays
n_p=60 ;number of points to use in finding the centroid
pix=findgen(n_el)   ;vector of all of the pixels
pix1=findgen(n_p+1)-n_p/2
pix2=findgen(31)-15
pix3=findgen(21)-10
pix4=findgen(15)-7
pix_fit=[findgen(10)-20,findgen(201)*.1-10,findgen(10)+11]
n_ideal=fltarr(n_shot)

offset=fltarr(n_el,n_shot) ;array of offsets for row
                           ;correction from fine data
offset_3=fltarr(n_el,n_shot)
offset_t=fltarr(n_el)
```

```
max_int=fltarr(n_el,n_shot)
max_i=fltarr(n_el,n_shot)
image_s=fltarr(n_p/2+1,n_el,n_shot)
x_s=fltarr(n_p+1,n_shot/3)
n_min=fltarr(n_chord,n_shot)
n_max=fltarr(n_chord,n_shot)

wave_set=[228.9,326.1,361.0,457.4,480.1,508.6, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          228.9,326.1,361.0,457.4,480.1,508.6, $
          643.5,652.3,686.0, $
          483.1,477.1,231.4,226.4 ] ;wave leangth for each shot
; ideal waveleangth for each shot
```

```
wave_i=[228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        228.8022,326.1055,361.0508,457.6044,479.992,508.5822, $
        643.847,650.5048,686.4066, $
        479.992 ,479.992 ,228.8022,228.8022]
port_i=[0,0,0,0,0,0, $
        0,0,0,0,0,0,0,0,0, $
        0,0,0,0,0,0,0,0,0, $
        0,0,0,0,0,0,0,0,0, $
        0,0,0,0,0,0, $
        0,0,0,0,0,0,0,0,0, $
        1,1,1,1,1,1, $
        1,1,1,1,1,1,1,1,1, $
        1,1,1,1,1,1,1,1,1, $
```

```
        1,1,1,1,1,1,1,1,1,1, $
        1,1,1,1,1,1, $
        1,1,1,1,1,1,1,1,1,1, $
        1,1,1,1]
view_i=port_i

grate_i=[0,0,0,0,0,0, $
         1,1,1,1,1,1,1,1,1,1, $
         2,2,2,2,2,2,2,2,2,2, $
         2,2,2,2,2,2,2,2,2,2, $
         0,0,0,0,0,0, $
         1,1,1,1,1,1,1,1,1,1, $
         0,0,0,0,0,0, $
         1,1,1,1,1,1,1,1,1,1, $
         2,2,2,2,2,2,2,2,2,2, $
         2,2,2,2,2,2,2,2,2,2, $
         0,0,0,0,0,0, $
         1,1,1,1,1,1,1,1,1,1, $
         1,1,0,0]    ;grating for each shot

for k=0,n_shot-1 do begin
  ;reading data
  ; changed to read_princeton on 11 April 01

  if ha_int ne 1 then begin
    read_princeton, 'd:\Zap Spec Data\ICCD_cal\18sep01\18sep01cal'+ $
                     stremo(k+1)+'.spe', $
                    image, x_calibration=wave
    ;
    ;saving images
```

```
  ;
  image_all(*,*,k)=image
  wave_all(*,k)=wave
endif else begin
  image=image_all(*,*,k)
  wave=wave_all(*,k)
endelse

;finding index of the ideal wavelength
n_ideal_temp=min(where(wave ge wave_i(k/3)))
n_ideal(k)=n_ideal_temp+(wave_i(k/3)-wave(n_ideal_temp))/ $
                        (wave(n_ideal_temp)-wave(n_ideal_temp-1))
x1=pix1+n_ideal(k) ;pixel range for ideal signal
x2=pix2+n_ideal(k) ;pixel range for ideal signal
x3=pix3+n_ideal(k) ;pixel range for ideal signal
x4=pix4+n_ideal(k) ;pixel range for ideal signal
pix1_t=pix1
pix2_t=pix2
if wave_i(k/3) gt 600. then begin
  x1=x2
  x2=x3
  pix1_t=pix2
  pix2_t=pix3
endif
window,0
plot,x1,image(x1,0),yrange=[0,1.1*max(image(x1,*))],$
     title='grate='+stremo(grate_i(k/3))+'   wave='+stremo(wave_i(k/3))
vline,n_ideal(k)
for i=0,n_el-1 do begin
  ;creating fine matrix for centriod calculation
```

```
;initial guess for center
max_int(i,k)=max(image(x1,i))
max_i(i,k)=min(where(image(x1,i) eq max_int(i,k)))
offset(i,k)=((-n_elements(x1)/2+max_i(i,k))>(-5.)<10)
offset_t(i)=offset(i,k)+1
count=0
while (abs(offset(i,k)-offset_t(i)) gt .05) and  $
      (count lt 20) do begin
  offset_t(i)=offset(i,k)
  image_c=interpol(image(*,i),pix-offset(i,k),x1)
  offset(i,k)=(((((int_tabulated(pix1_t,image_c*pix1_t)/ $
               int_tabulated(pix1_t,image_c))<1)>(-1))+ $
               offset_t(i))>(-5.)<10)
  count=count+1
endwhile

oplot,x1,image_c,color=(count+1)>2
offset_t(i)=offset(i,k)+1
count=0
while (abs(offset(i,k)-offset_t(i)) gt .05) and $
      (count lt 20) do begin
  offset_t(i)=offset(i,k)
  image_c=interpol(image(*,i),pix-offset(i,k),x2)
  offset(i,k)=(((((int_tabulated(pix2_t,image_c*pix2_t)/ $
              int_tabulated(pix2_t,image_c))<1)>(-1))+ $
              offset_t(i))>(-5.)<10)
  count=count+1
endwhile
offset_t(i)=offset(i,k)+1
count=0
```

```
  while (abs(offset(i,k)-offset_t(i)) gt .05) and (count lt 20) do begin
    offset_t(i)=offset(i,k)
    image_c=interpol(image(*,i),pix-offset(i,k),x3)
    offset(i,k)=(((((int_tabulated(pix3,image_c*pix3)/ $
                  int_tabulated(pix3,image_c))<1)>(-1))+ $
                  offset_t(i))>(-5.)<10)
    count=count+1
  endwhile
  oplot,x3,image_c,color=(count+1)>2,linestyle=1
  offset_3(i,k)=offset(i,k)
  offset_t(i)=offset(i,k)+1
  count=0

  while (abs(offset(i,k)-offset_t(i)) gt .05) and (count lt 20) do begin
    offset_t(i)=offset(i,k)
    image_c=interpol(image(*,i),pix-offset(i,k),x4)
    offset(i,k)=(((((int_tabulated(pix4,image_c*pix4)/ $
                  int_tabulated(pix4,image_c))<1)>(-1))+ $
                  offset_t(i))>(-5.)<10)
    count=count+1
  endwhile
 oplot,x4,image_c,color=20*i/n_el+2,linestyle=2
 max_int(i,k)=max(image_c)
endfor

max_sm=smooth(smooth(max_int(*,k),11,/EDGE_TRUNCATE),7,/EDGE_TRUNCATE)
a=(max_sm(0:n_el-2)-max_sm(1:n_el-1)) lt 1
b=a(0:509)-a(1:510)
c=where(b ne 0)
```

```
n_el_c=n_elements(c)
if n_el_c lt (n_chord*2-1) then begin
   c_temp=c
   c=fltarr(n_chord*2-1)
   c(1:n_el_c)=c_temp
endif

for j=0,n_chord-1 do begin
   l_m=c(2*j)+1
   if j ne 0 then  $
      find_stop=c(2*j-1)+1 else  $
      find_stop=0
   n_min(j,k)=l_m
   while (max_sm((n_min(j,k)-1)>0) gt  $
         (r_level*(max_sm(l_m)-max_sm(find_stop))+max_sm(find_stop))) $
         and (n_min(j,k) gt find_stop) do $
         n_min(j,k)=n_min(j,k)-1
   if j ne n_chord-1 then  $
      find_stop=c(2*j+1)+1 else  $
      find_stop=n_el-1
   n_max(j,k)=l_m
   while (max_sm((n_max(j,k)+1)<(n_el-1)) gt $
         (r_level*(max_sm(l_m)-max_sm(find_stop))+max_sm(find_stop))) $
         and (n_max(j,k) lt find_stop) do $
         n_max(j,k)=n_max(j,k)+1

endfor
window,1,xsize=500,ysize=400,xpos=0,ypos=100,title='Window 1'
loadct,10
shade_surf,image_all(x1,*,k),ax=90,az=0;,yrange=[0,100]
```

```
  tek_color
  contour,image_all(x1,*,k),/noerase,nlevel=3,title='k='+stremo(k)
  for i=0,n_chord-1 do begin
     hline,n_min(i,k)
     hline,n_max(i,k)
  endfor
  oplot,offset(*,k)+n_elements(x1)/2,pix
endfor

tek_color

bin1:
SET_PLOT, 'PS'
DEVICE, FILENAME= $
          'd:\Zap Spec Data\ICCD_cal\18sep01\18sep01_check_600.ps', $
          /portrait, /color, $
          ysize=25,yoffset=1.2
!p.multi=[0,1,3]

for k1=0,n_shot/3-1 do begin
plot,offset(*,3*k1), $
     title='offset cal='+stremo(3*k1+1)+'  wave='+stremo(wave_i(k1))
for k=1,2 do $
    oplot,offset(*,3*k1+k),color=k+1
plot,n_min(*,3*k1), $
     title='n_min cal='+stremo(3*k1+1)+'  Grate='+stremo(grate_i(k1))
for k=1,2 do $
    oplot,n_min(*,3*k1+k),color=k+1
plot,n_max(*,3*k1),title='n_max cal='+stremo(3*k+1)
for k=1,2 do $
```

```
    oplot,n_max(*,3*k1+k),color=k+1
endfor

DEVICE, /CLOSE
set_plot,'win'


SET_PLOT, 'PS'
DEVICE, FILENAME= $
        'd:\Zap Spec Data\ICCD_cal\18sep01\18sep01_cal_var1_600.ps', $
        /landscape, /color
!p.multi=[0,3,1]
for k=0,n_shot-1 do begin
  x1=pix1+n_ideal(k)
  contour,image_all(x1,*,k),nlevel=10,title='raw cal='+ $
          stremo(k+1),c_colors=indgen(5)+2, $
          xrange=[20,40];,yrange=[0,100]
  for i=0,n_chord-1 do begin
     hline,n_min(i,k),color=2
     hline,n_max(i,k),color=4
  endfor
  oplot,offset(*,k)+n_elements(x1)/2,pix,color=2
endfor

;finding row offsets and bins to use

row_offset=fltarr(n_el,n_shot/3)
bin_min=fltarr(n_chord,n_shot/3)
bin_max=fltarr(n_chord,n_shot/3)
err_row_offset=fltarr(n_el,n_shot/3)
```

```
err_bin_min=fltarr(n_chord,n_shot/3)
err_bin_max=fltarr(n_chord,n_shot/3)
image_shift=fltarr(n_el,n_el)
image_bin=fltarr(n_el,n_chord,n_shot)
d_image_bin=fltarr(n_chord,n_shot)
area_fine=fltarr(n_chord,n_shot)
area_coarse=fltarr(n_chord,n_shot)
fwhm_s=fltarr(n_chord,n_shot)
ratio_shot=fltarr(n_chord,n_shot)
inst_line_s=fltarr(n_p+1,n_chord,n_shot)
row_offset_3=fltarr(n_el,n_shot/3)
err_row_offset_3=fltarr(n_el,n_shot/3)

!p.multi=[0,4,5]

for k=0,n_shot/3-1 do begin
   row_offset(*,k)=(offset(*,3*k)+offset(*,3*k+1)+offset(*,3*k+2))/3
   err_row_offset(*,k)=sqrt((offset(*,3*k)-row_offset(*,k))^2 + $
                            (offset(*,3*k+1)-row_offset(*,k))^2 + $
                            (offset(*,3*k+2)-row_offset(*,k))^2)/3
   row_offset_3(*,k)=(offset_3(*,3*k)+ $
                      offset_3(*,3*k+1)+offset_3(*,3*k+2))/3
   err_row_offset_3(*,k)=sqrt((offset_3(*,3*k)-row_offset_3(*,k))^2 + $
                            (offset_3(*,3*k+1)-row_offset_3(*,k))^2 + $
                            (offset_3(*,3*k+2)-row_offset_3(*,k))^2)/3
   for i=0,n_chord-1 do begin
      bin_min(i,k)=max(n_min(i,3*k:(3*k+2)))
      bin_max(i,k)=min(n_max(i,3*k:(3*k+2)))
   endfor
   err_bin_min(*,k)=sqrt((n_min(*,3*k)-bin_min(*,k))^2 + $
```

```
                              (n_min(*,3*k+1)-bin_min(*,k))^2 + $
                              (n_min(*,3*k+2)-bin_min(*,k))^2)
   err_bin_max(*,k)=sqrt((n_max(*,3*k)-bin_max(*,k))^2 + $
                              (n_max(*,3*k+1)-bin_max(*,k))^2 + $
                              (n_max(*,3*k+2)-bin_max(*,k))^2)
   plot,row_offset(*,k)
   oplot,row_offset(*,k)+err_row_offset(*,k),linestyle=2
   oplot,row_offset(*,k)-err_row_offset(*,k),linestyle=2
   oplot,row_offset_3(*,k),color=2
   oplot,row_offset_3(*,k)+err_row_offset_3(*,k),color=2,linestyle=2
   oplot,row_offset_3(*,k)-err_row_offset_3(*,k),color=2,linestyle=2

endfor
bin_min(0,*)=0
;finding correction parameters for each shot

!p.multi=[0,3,1]

for k=0,n_shot-1 do begin
   ;aligning rows
   for i=0,n_el-1 do begin
    image_shift(*,i)=interpol(image_all(*,i,k),pix-row_offset(i,k/3),pix)
   endfor
   ;window,1,xsize=1000,ysize=750,xpos=0,ypos=100,title='Window 1'
   tek_color
   x1=pix1+n_ideal(k)
   contour,image_shift(x1,*),nlevel=10,title='cal'+stremo(k+1), $
           c_colors=indgen(5)+2, $
           xrange=[20,40];,yrange=[0,100]
   for i=0,n_chord-1 do begin
```

```
      hline,n_min(i,k),color=2
      hline,n_max(i,k),color=4
   endfor

   ;binning data
   for j=0,n_chord-1 do begin

      bin_max=bin_max>bin_min
      for i=0,n_el-1 do $
         image_bin(i,j,k)= $
           total(image_shift(i,bin_min(j,k/3):bin_max(j,k/3)))
      x=pix1+n_ideal(k)
      small_x=pix_fit+n_ideal(k)
      y=image_bin(x,j,k)
      y_f=spline(x,y,small_x,1)
      area_fine(j,k)=int_tabulated(small_x,y_f)
      area_coarse(j,k)=total(y)

      max_bin=max(y_f)
      FWHM_s(j,k)=small_x((max(where(y_f gt .5*max_bin))<n_p*10)>0) - $
                small_x(min(where(y_f gt .5*max_bin))>0)
      inst_line_s(*,j,k)=y
   endfor
   ;finding correction factors for each shot
   ratio_shot(*,k)=area_coarse(10,k)/area_coarse(*,k)
endfor

ratio=fltarr(n_chord,n_shot/3)
fwhm=fltarr(n_chord,n_shot/3)
inst_line=fltarr(n_p+1,n_chord,n_shot/3)
```

```
err_ratio=fltarr(n_chord,n_shot/3)
err_fwhm=fltarr(n_chord,n_shot/3)
err_inst_line=fltarr(n_p+1,n_chord,n_shot/3)
area=fltarr(n_chord,n_shot/3)
err_area=fltarr(n_chord,n_shot/3)

;finding correction factors
for k=0,n_shot/3-1 do begin
   area(*,k)=(area_coarse(*,3*k)+ $
              area_coarse(*,3*k+1)+area_coarse(*,3*k+2))/3
   err_area(*,k)=sqrt((area_coarse(*,3*k)-area(*,k))^2 + $
                      (area_coarse(*,3*k+1)-area(*,k))^2 + $
                      (area_coarse(*,3*k+2)-area(*,k))^2)/3
   fwhm(*,k)=(fwhm_s(*,3*k)+fwhm_s(*,3*k+1)+fwhm_s(*,3*k+2))/3
   ratio(*,k)=(ratio_shot(*,3*k)+ratio_shot(*,3*k+1)+ratio_shot(*,3*k+2))/3
   err_ratio(*,k)=sqrt((ratio_shot(*,3*k)-ratio(*,k))^2 + $
                      (ratio_shot(*,3*k+1)-ratio(*,k))^2 + $
                      (ratio_shot(*,3*k+2)-ratio(*,k))^2)/3
   fwhm(*,k)=(fwhm_s(*,3*k)+fwhm_s(*,3*k+1)+fwhm_s(*,3*k+2))/3
   err_fwhm(*,k)=sqrt((fwhm_s(*,3*k)-fwhm(*,k))^2 + $
                     (fwhm_s(*,3*k+1)-fwhm(*,k))^2 + $
                     (fwhm_s(*,3*k+2)-fwhm(*,k))^2)/3
   for j=0,n_chord-1 do $
      inst_line(*,j,k)=(inst_line_s(*,j,3*k)/area_coarse(j,3*k)+ $
                        inst_line_s(*,j,3*k+1)/area_coarse(j,3*k+1)+ $
                        inst_line_s(*,j,3*k+2)/area_coarse(j,3*k+2))/3
   !p.multi=[0,2,2]
   plot,inst_line(*,0,k),title='shot='+stremo(k*3+1)
   for i=1,4 do oplot,inst_line(*,0+i,k),color=i+1
   plot,inst_line(*,5,k),title='shot='+stremo(k*3+1)
```

```
   for i=1,4 do oplot,inst_line(*,5+i,k),color=i+1
   plot,inst_line(*,10,k),title='shot='+stremo(k*3+1)
   for i=1,4 do oplot,inst_line(*,10+i,k),color=i+1
   plot,inst_line(*,15,k),title='shot='+stremo(k*3+1)
   for i=1,4 do oplot,inst_line(*,15+i,k),color=i+1
   !p.multi=0
endfor

DEVICE, /CLOSE
set_plot,'win'

if n_shot eq 300 then save,FILENAME =
  'd:\Zap Spec Data\ICCD_cal\18sep01\18sep01_cal_var2_600.dat',/all

end
```

save_cal_18sep01_param.pro

```
;pro save_cal_18sep01_param

;All parameters are a function of wavelength, view, and grating
; Wavelength
; 0->min(wave)<275
; 1->275<min(wave)<340
; 2->340<min(wave)<420
; 3->420<min(wave)<520
; 4->520<min(wave)<600
; 5->600<min(wave)<675
; 6->675<min(wave)

; should use 277.454,343.578,420.521,494.287,576.215,665.127
;  for this calibration
; will use 275,340,420,495,575,665 for this calibration

; Entrance port
; 0->front
; 1->side
; Grating
; 2->150
; 1->2400
; 0->3600

default,have,1
default,ps,0

if have ne 0 then begin
```

```
  restore,FILENAME = 'j:\Zap Spec Data\ICCD_cal\18sep01\' + $
                     '18sep01_cal_var2_600.dat'
  !p.position=0
  !p.multi=[0,2,4]

  row_offset_t=row_offset(*,0:95)
  err_row_offset_t=err_row_offset(*,0:95)
  bin_min_t=bin_min(*,0:95)
  bin_max_t=bin_max(*,0:95)
  rel_i_t=ratio(*,0:95)
  err_rel_i_t=err_ratio(*,0:95)
  wave_i=wave_i(0:95)

  ;averaging parameters
  wave_test=[228.8022,326.1055,361.0508,479.992,508.5822,643.847,686.4066]
  n_wt=n_elements(wave_test)

  ;initializing arrays
  row_offset=fltarr(512,7,2,3)
  bin_min=fltarr(20,7,2,3)
  bin_max=fltarr(20,7,2,3)
  rel_i=fltarr(20,7,2,3)
  fwhm_inst=fltarr(20,7,2,3)
  inst_func=fltarr(61,20,7,2,3)
  wav_i=fltarr(96)
endif

for i=0,95 do begin
   if wave(n_el/2) le 275 then wav_i=0 $
     else if wave(n_el/2) le 340 then wav_i=1 $
```

```
      else if wave(n_el/2) le 420 then wav_i=2 $
      else if wave(n_el/2) le 495 then wav_i=3 $
      else if wave(n_el/2) le 575 then wav_i=4 $
      else if wave(n_el/2) le 665 then wav_i=5 $
      else wav_i=6
   ;row_offset(*,wav_i(i),view_i(i),(grate_i(i)))=row_offset_t(*,i)
   ;bin_min(*,wav_i(i),view_i(i),(grate_i(i)))=bin_min_t(*,i)
   ;bin_max(*,wav_i(i),view_i(i),(grate_i(i)))=bin_max_t(*,i)
   ;rel_i(*,wav_i(i),view_i(i),(grate_i(i)))=rel_i_t(*,i)
   ;fwhm_inst(*,wav_i(i),view_i(i),(grate_i(i)))=fwhm(*,i)
   ;inst_func(*,*,wav_i(i),view_i(i),(grate_i(i)))=inst_line(*,*,i)
endfor

for i=0,n_wt-1 do begin

   n_wave_test=where(wave_i eq wave_test(i))
   n_nwt=n_elements(wave_test)
   n_36=where(grate_i(n_wave_test) eq 0)
   n_24=where(grate_i(n_wave_test) eq 1)
   n_15=where(grate_i(n_wave_test) eq 2)
   if min(n_36) ne -1 then begin
     n_front=where(view_i(n_wave_test(n_36)) eq 0)
     row_offset(*,i,0,0)= $
               (row_offset_t(*,(n_wave_test(n_36(n_front(0)))))+ $
                row_offset_t(*,(n_wave_test(n_36(n_front(1))))))/2
     bin_min(*,i,0,0)= $
               (bin_min_t(*,(n_wave_test(n_36(n_front(0)))))+ $
                bin_min_t(*,(n_wave_test(n_36(n_front(1))))))/2
     bin_max(*,i,0,0)= $
               (bin_max_t(*,(n_wave_test(n_36(n_front(0)))))+ $
```

```
           bin_max_t(*,(n_wave_test(n_36(n_front(1))))))/2
rel_i(*,i,0,0)= $
          (rel_i_t(*,(n_wave_test(n_36(n_front(0)))))+ $
           rel_i_t(*,(n_wave_test(n_36(n_front(1))))))/2
fwhm_inst(*,i,0,0)= $
          (fwhm(*,(n_wave_test(n_36(n_front(0)))))+ $
           fwhm(*,(n_wave_test(n_36(n_front(1))))))/2
inst_func(*,*,i,0,0)= $
          (inst_line(*,*,(n_wave_test(n_36(n_front(0)))))+ $
           inst_line(*,*,(n_wave_test(n_36(n_front(1))))))/2

n_side=where(view_i(n_wave_test(n_36)) eq 1)
row_offset(*,i,1,0)= $
          (row_offset_t(*,(n_wave_test(n_36(n_side(0)))))+ $
           row_offset_t(*,(n_wave_test(n_36(n_side(1))))))/2
bin_min(*,i,1,0)= $
          (bin_min_t(*,(n_wave_test(n_36(n_side(0)))))+ $
           bin_min_t(*,(n_wave_test(n_36(n_side(1))))))/2
bin_max(*,i,1,0)= $
          (bin_max_t(*,(n_wave_test(n_36(n_side(0)))))+ $
           bin_max_t(*,(n_wave_test(n_36(n_side(1))))))/2
rel_i(*,i,1,0)= $
          (rel_i_t(*,(n_wave_test(n_36(n_side(0)))))+ $
           rel_i_t(*,(n_wave_test(n_36(n_side(1))))))/2
fwhm_inst(*,i,1,0)= $
          (fwhm(*,(n_wave_test(n_36(n_side(0)))))+ $
           fwhm(*,(n_wave_test(n_36(n_side(1))))))/2
inst_func(*,*,i,1,0)= $
          (inst_line(*,*,(n_wave_test(n_36(n_side(0)))))+ $
           inst_line(*,*,(n_wave_test(n_36(n_side(1))))))/2
```

```
endif
  n_front=where(view_i(n_wave_test(n_24)) eq 0)
  row_offset(*,i,0,1)= $
            (row_offset_t(*,(n_wave_test(n_24(n_front(0)))))+ $
             row_offset_t(*,(n_wave_test(n_24(n_front(1))))))/2
  bin_min(*,i,0,1)= $
            (bin_min_t(*,(n_wave_test(n_24(n_front(0)))))+ $
             bin_min_t(*,(n_wave_test(n_24(n_front(1))))))/2
  bin_max(*,i,0,1)= $
            (bin_max_t(*,(n_wave_test(n_24(n_front(0)))))+ $
             bin_max_t(*,(n_wave_test(n_24(n_front(1))))))/2
  rel_i(*,i,0,1)= $
            (rel_i_t(*,(n_wave_test(n_24(n_front(0)))))+ $
             rel_i_t(*,(n_wave_test(n_24(n_front(1))))))/2
  fwhm_inst(*,i,0,1)= $
            (fwhm(*,(n_wave_test(n_24(n_front(0)))))+ $
             fwhm(*,(n_wave_test(n_24(n_front(1))))))/2
  inst_func(*,*,i,0,1)= $
            (inst_line(*,*,(n_wave_test(n_24(n_front(0)))))+ $
             inst_line(*,*,(n_wave_test(n_24(n_front(1))))))/2

  n_side=where(view_i(n_wave_test(n_24)) eq 1)
  row_offset(*,i,1,1)= $
            (row_offset_t(*,(n_wave_test(n_24(n_side(0)))))+ $
             row_offset_t(*,(n_wave_test(n_24(n_side(1))))))/2
  bin_min(*,i,1,1)= $
            (bin_min_t(*,(n_wave_test(n_24(n_side(0)))))+ $
             bin_min_t(*,(n_wave_test(n_24(n_side(1))))))/2
  bin_max(*,i,1,1)= $
            (bin_max_t(*,(n_wave_test(n_24(n_side(0)))))+ $
```

```
           bin_max_t(*,(n_wave_test(n_24(n_side(1))))))/2
rel_i(*,i,1,1)= $
          (rel_i_t(*,(n_wave_test(n_24(n_side(0)))))+ $
           rel_i_t(*,(n_wave_test(n_24(n_side(1))))))/2
fwhm_inst(*,i,1,1)= $
          (fwhm(*,(n_wave_test(n_24(n_side(0)))))+ $
           fwhm(*,(n_wave_test(n_24(n_side(1))))))/2
inst_func(*,*,i,1,1)= $
          (inst_line(*,*,(n_wave_test(n_24(n_side(0)))))+ $
           inst_line(*,*,(n_wave_test(n_24(n_side(1))))))/2
n_front=where(view_i(n_wave_test(n_15)) eq 0)
row_offset(*,i,0,2)= $
          (row_offset_t(*,(n_wave_test(n_15(n_front(0)))))+ $
           row_offset_t(*,(n_wave_test(n_15(n_front(1))))))/2
bin_min(*,i,0,2)= $
          (bin_min_t(*,(n_wave_test(n_15(n_front(0)))))+ $
           bin_min_t(*,(n_wave_test(n_15(n_front(1))))))/2
bin_max(*,i,0,2)= $
          (bin_max_t(*,(n_wave_test(n_15(n_front(0)))))+ $
           bin_max_t(*,(n_wave_test(n_15(n_front(1))))))/2
rel_i(*,i,0,2)= $
          (rel_i_t(*,(n_wave_test(n_15(n_front(0)))))+ $
           rel_i_t(*,(n_wave_test(n_15(n_front(1))))))/2
fwhm_inst(*,i,0,2)= $
          (fwhm(*,(n_wave_test(n_15(n_front(0)))))+ $
           fwhm(*,(n_wave_test(n_15(n_front(1))))))/2
inst_func(*,*,i,0,2)= $
          (inst_line(*,*,(n_wave_test(n_15(n_front(0)))))+ $
           inst_line(*,*,(n_wave_test(n_15(n_front(1))))))/2
```

```
     n_side=where(view_i(n_wave_test(n_15)) eq 1)
     row_offset(*,i,1,2)= $
               (row_offset_t(*,(n_wave_test(n_15(n_side(0)))))+ $
                row_offset_t(*,(n_wave_test(n_15(n_side(1))))))/2
     bin_min(*,i,1,2)= $
               (bin_min_t(*,(n_wave_test(n_15(n_side(0)))))+ $
                bin_min_t(*,(n_wave_test(n_15(n_side(1))))))/2
     bin_max(*,i,1,2)= $
               (bin_max_t(*,(n_wave_test(n_15(n_side(0)))))+ $
                bin_max_t(*,(n_wave_test(n_15(n_side(1))))))/2
     rel_i(*,i,1,2)= $
               (rel_i_t(*,(n_wave_test(n_15(n_side(0)))))+ $
                rel_i_t(*,(n_wave_test(n_15(n_side(1))))))/2
     fwhm_inst(*,i,1,2)= $
               (fwhm(*,(n_wave_test(n_15(n_side(0)))))+ $
                fwhm(*,(n_wave_test(n_15(n_side(1))))))/2
     inst_func(*,*,i,1,2)= $
               (inst_line(*,*,(n_wave_test(n_15(n_side(0)))))+ $
                inst_line(*,*,(n_wave_test(n_15(n_side(1))))))/2

endfor

save,row_offset,bin_min,bin_max,rel_i,fwhm_inst,inst_func, $
     FILENAME ='j:\Zap Spec Data\ICCD_cal\18sep01\' $
               'iccd_cal_param_18sep01.dat'
row_offset_new=row_offset
bin_min_new=bin_min
bin_max_new=bin_max
rel_i_new=rel_i
fwhm_inst_new=fwhm_inst
```

```
inst_func_new=inst_func
restore,'j:\Zap Spec Data\iccd_data\iccd_cal_param.dat'

if ps eq 1 then begin
  SET_PLOT, 'PS'
  DEVICE, FILENAME='j:\Zap Spec Data\ICCD_cal\18sep01\' = $
          '18sep01_check_cal_factors.ps', $
          /portrait, /color, $
          ysize=25,yoffset=1.2
endif

!p.multi=[0,2,5]
winct=0

for i=0,2 do begin
   !p.multi=[0,2,n_wt]
   if ps ne 1 then window,i*2
   for j=0,n_wt-1 do begin
      plot,row_offset_new(*,j,0,i), $
           yrange=[-10,10], $
           title='grate='+stremo(i)+'wave='+stremo(wave_test(j))
      oplot,row_offset(*,j,0,i),color=2,linestyle=2
      plot,row_offset_new(*,j,1,i), $
           yrange=[min(row_offset_new(*,*,1,i)) $
           ,max(row_offset_new(*,*,1,i))], $
           title='grate='+stremo(i)+'wave='+stremo(wave_test(j))
      oplot,row_offset(*,j,1,i),color=2,linestyle=2
   endfor
   !p.multi=[0,2,2]
   if ps ne 1 then window,i*2+1
```

```
plot,wave_test,transpose(bin_min_new(0,*,0,i)), $
     yrange=[0,512]
for j=0,19 do oplot,wave_test,transpose(bin_min_new(j,*,0,i)), $
                   color=2+j
for j=0,19 do oplot,wave_test,transpose(bin_min(j,*,0,i)), $
                   color=2+j,linestyle=2
plot,wave_test,transpose(bin_min_new(0,*,1,i)), $
     yrange=[0,512]
for j=0,19 do oplot,wave_test,transpose(bin_min_new(j,*,1,i)), $
                   color=2+j
for j=0,19 do oplot,wave_test,transpose(bin_min(j,*,1,i)), $
                   color=2+j,linestyle=2
plot,wave_test,transpose(bin_max_new(0,*,0,i)), $
     yrange=[0,512]
for j=0,19 do oplot,wave_test,transpose(bin_max_new(j,*,0,i)), $
                   color=2+j
for j=0,19 do oplot,wave_test,transpose(bin_max(j,*,0,i)), $
                   color=2+j,linestyle=2
plot,wave_test,transpose(bin_max_new(0,*,1,i)), $
     yrange=[0,512]
for j=0,19 do oplot,wave_test,transpose(bin_max_new(j,*,1,i)), $
                   color=2+j
for j=0,19 do oplot,wave_test,transpose(bin_max(j,*,1,i)), $
                   color=2+j,linestyle=2
plot,wave_test,transpose(rel_i_new(0,*,0,i)), $
     yrange=[0,2]
for j=0,19 do oplot,wave_test,transpose(rel_i_new(j,*,0,i)), $
                   color=2+j
for j=0,19 do oplot,wave_test,transpose(rel_i(j,*,0,i)), $
                   color=2+j,linestyle=2
```

```
   plot,wave_test,transpose(rel_i_new(0,*,1,i)), $
        yrange=[0,2]
   for j=0,19 do oplot,wave_test,transpose(rel_i_new(j,*,1,i)), $
                       color=2+j
   for j=0,19 do oplot,wave_test,transpose(rel_i(j,*,1,i)), $
                       color=2+j,linestyle=2
   plot,wave_test,transpose(fwhm_inst_new(0,*,0,i)), $
        yrange=[2,7]
   for j=0,19 do oplot,wave_test,transpose(fwhm_inst_new(j,*,0,i)), $
                       color=2+j
   for j=0,19 do oplot,wave_test,transpose(fwhm_inst(j,*,0,i)), $
                       color=2+j,linestyle=2
   plot,wave_test,transpose(fwhm_inst_new(0,*,1,i)), $
        yrange=[2,7]
   for j=0,19 do oplot,wave_test,transpose(fwhm_inst_new(j,*,1,i)), $
                       color=2+j
   for j=0,19 do oplot,wave_test,transpose(fwhm_inst(j,*,1,i)), $
                       color=2+j,linestyle=2
   ;plot,wave_test,transpose(inst_func(*,*,i,0,0))
endfor

!p.multi=0

if ps eq 1 then begin
  DEVICE, /CLOSE
  set_plot,'win'
endif

end
```

C.4.2 fitting routines and functions

The codes in this section are used to fit Gaussians or instrument widened Gaussians to the measured spectra. *err_CURVEFIT_wide.pro* is a modified version of the IDL routine *CURVEFIT* which returns a flag when errors are encountered and values used to calculate the correct errors. The other functions in this section are examples of the different fitting functions which are used to model the spectral intensities. *fgauss3.pro* is the function used when fitting the amplitude of a Gaussian with an offset. *f_emiss_wide.pro* is the function used when fitting the amplitude of a Gaussian with an offset. When a doublet or triplet is being fit, the difference of central wavelengths, the relative intensity and the FWHM of each Gaussian is fixed. *f_emiss_wide_three_cV_227* is the function used when fitting the area of multiple Gaussians, which have been widened by the instrument function including an offset.

err_CURVEFIT_wide.pro

```
; $Id: curvefit.pro,v 1.21 1999/01/16 01:22:27 scottm Exp $
;
; Copyright (c) 1982-1999, Research Systems, Inc.  All rights reserved.
; Unauthorized reproduction prohibited.
;
FUNCTION err_CURVEFIT_wide, x, y, weights, a, a1, sigma, $
                        FUNCTION_NAME = Function_Name, $
                        ITMAX=itmax, ITER=iter, TOL=tol, CHI2=chi2, $
                        NODERIVATIVE=noderivative, CHISQ=chisq, $
                        curve_err=curve_err,pder=pder
;
;This is a modification of the curve fit program.
;It returns an a constant "curve_err=0" if the curve fit failed
;If the fit succeded then curve_err=1
;+
; NAME:
;       CURVEFIT
;
; PURPOSE:
;       Non-linear least squares fit to a function of an arbitrary
;       number of parameters.  The function may be any non-linear
;       function.  If available, partial derivatives can be calculated by
;       the user function, else this routine will estimate partial
;       derivatives with a forward difference approximation.
;
; CATEGORY:
;       E2 - Curve and Surface Fitting.
;
```

```
; CALLING SEQUENCE:
;       Result = CURVEFIT(X, Y, Weights, A, SIGMA, FUNCTION_NAME = name, $
;                         ITMAX=ITMAX, ITER=ITER, TOL=TOL, /NODERIVATIVE)
;
; INPUTS:
;       X:  A row vector of independent variables.  This routine does
;           not manipulate or use values in X, it simply passes X
;           to the user-written function.
;
;       Y:  A row vector containing the dependent variable.
;
;  Weights:  A row vector of weights, the same length as Y.
;            For no weighting,
;                 Weights(i) = 1.0.
;            For instrumental (Gaussian) weighting,
;                 Weights(i)=1.0/sigma(i)^2
;            For statistical (Poisson)  weighting,
;                 Weights(i) = 1.0/y(i), etc.
;
;       A:  A vector, with as many elements as the number of terms, that
;           contains the initial estimate for each parameter.  IF A is
;           double precision, calculations are performed in double
;           precision, otherwise they are performed in single precision.
;           Fitted parameters are returned in A.
;
; KEYWORDS:
;       FUNCTION_NAME:  The name of the function (actually, a procedure) to
;       fit.  IF omitted, "FUNCT" is used. The procedure must be written as
;       described under RESTRICTIONS, below.
;
```

```
;       ITMAX:  Maximum number of iterations. Default = 20.
;       ITER:   The actual number of iterations which were performed
;       TOL:    The convergence tolerance. The routine returns when the
;               relative decrease in chi-squared is less than TOL in an
;               interation. Default = 1.e-3.
;       CHI2:   The value of chi-squared on exit (obselete)
;
;       CHISQ:   The value of reduced chi-squared on exit
;       NODERIVATIVE:   IF this keyword is set THEN the user procedure
;               will not be requested to provide partial derivatives.
;               The partial derivatives will be estimated in CURVEFIT
;               using forward differences. IF analytical derivatives
;               are available they should always be used.
;
; OUTPUTS:
;       Returns a vector of calculated values.
;       A:  A vector of parameters containing fit.
;
; OPTIONAL OUTPUT PARAMETERS:
;       Sigma:  A vector of standard deviations for the parameters in A.
;
; COMMON BLOCKS:
;       NONE.
;
; SIDE EFFECTS:
;       None.
;
; RESTRICTIONS:
;       The function to be fit must be defined and called FUNCT,
;       unless the FUNCTION_NAME keyword is supplied.  This function,
```

```
;       (actually written as a procedure) must accept values of
;       X (the independent variable), and A (the fitted function's
;       parameter values), and return F (the function's value at
;       X), and PDER (a 2D array of partial derivatives).
;       For an example, see FUNCT in the IDL User's Libaray.
;       A call to FUNCT is entered as:
;       FUNCT, X, A, F, PDER
; where:
;       X = Variable passed into CURVEFIT.  It is the job of the
;           user-written function to interpret this variable.
;       A = Vector of NTERMS function parameters, input.
;       F = Vector of NPOINT values of function, y(i) = funct(x), output.
;       PDER = Array, (NPOINT, NTERMS), of partial derivatives of funct.
;               PDER(I,J) = DErivative of function at ith point with
;               respect to jth parameter.  Optional output parameter.
;               PDER should not be calculated IF the parameter is not
;               supplied in call. IF the /NODERIVATIVE keyword is set in
;               the call to CURVEFIT THEN the user routine will never need
;               to calculate PDER.
;
; PROCEDURE:
;       Copied from "CURFIT", least squares fit to a non-linear
;       function, pages 237-239, Bevington, Data Reduction and Error
;       Analysis for the Physical Sciences.  This is adapted from:
;       Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear
;       Parameters", J. Soc. Ind. Appl. Math., Vol 11, no. 2, pp. 431-441,
;       June, 1963.
;
;       "This method is the Gradient-expansion algorithm which
;       combines the best features of the gradient search with
```

```
;       the method of linearizing the fitting function."
;
;       Iterations are performed until the chi square changes by
;       only TOL or until ITMAX iterations have been performed.
;
;       The initial guess of the parameter values should be
;       as close to the actual values as possible or the solution
;       may not converge.
;
; EXAMPLE:  Fit a function of the form f(x) = a * exp(b*x) + c to
;           sample pairs contained in x and y.
;           In this example, a=a(0), b=a(1) and c=a(2).
;           The partials are easily computed symbolicaly:
;           df/da = exp(b*x), df/db = a * x * exp(b*x), and df/dc = 1.0
;
;           Here is the user-written procedure to return F(x) and
;           the partials, given x:
;
;       pro gfunct, x, a, f, pder      ; Function + partials
;         bx = exp(a(1) * x)
;         f= a(0) * bx + a(2)         ;Evaluate the function
;         IF N_PARAMS() ge 4 THEN $   ;Return partials?
;         pder= [[bx], [a(0) * x * bx], [replicate(1.0, N_ELEMENTS(f))]]
;       end
;
;         x=findgen(10)                  ;Define indep & dep variables.
;         y=[12.0, 11.0,10.2,9.4,8.7,8.1,7.5,6.9,6.5,6.1]
;         Weights=1.0/y            ;Weights
;         a=[10.0,-0.1,2.0]        ;Initial guess
;         yfit=curvefit(x,y,Weights,a,sigma,function_name='gfunct')
```

```
;         print, 'Function parameters: ', a
;         print, yfit
;       end
;
; MODIFICATION HISTORY:
;       Written, DMS, RSI, September, 1982.
;       Does not iterate IF the first guess is good.  DMS, Oct, 1990.
;       Added CALL_PROCEDURE to make the function's name a parameter.
;              (Nov 1990)
;       12/14/92 - modified to reflect the changes in the 1991
;            edition of Bevington (eq. II-27) (jiy-suggested by CreaSo)
;       Mark Rivers, U of Chicago, Feb. 12, 1995
;           - Added following keywords: ITMAX, ITER, TOL, CHI2,
;             NODERIVATIVE
;             These make the routine much more generally useful.
;           - Removed Oct. 1990 modification so the routine does one
;             iteration even IF first guess is good. Required to get
;             meaningful output for errors.
;           - Added forward difference derivative calculations required for
;             NODERIVATIVE keyword.
;           - Fixed a bug: PDER was passed to user's procedure on first
;             call, but was not defined. Thus, user's procedure might
;             not calculate it, but the result was THEN used.
;
;      Steve Penton, RSI, June 1996.
;            - Changed SIGMAA to SIGMA to be consistant with other fitting
;              routines.
;            - Changed CHI2 to CHISQ to be consistant with other fitting
;              routines.
;            - Changed W to Weights to be consistant with other fitting
```

```
;               routines.
;             _ Updated docs regarding weighing.
;
;-
        ON_ERROR,2              ;Return to caller IF error
        curve_err=1             ;Set intial error to 1 (no error)

        ;Name of function to fit

        IF n_elements(function_name) LE 0 THEN function_name = "FUNCT"

        IF n_elements(tol) EQ 0 THEN tol = 1.e-3      ;Convergence tolerance
        IF n_elements(itmax) EQ 0 THEN itmax = 20     ;Maximum # iterations
        type = size(a)
        type = type[type[0]+1]
        double = type EQ 5

        ;Make params floating
        IF (type ne 4) AND (type ne 5) THEN a = float(a)

        ; IF we will be estimating partial derivatives THEN compute machine
        ; precision

        IF keyword_set(NODERIVATIVE) THEN BEGIN
           res = machar(DOUBLE=double)
           eps = sqrt(res.eps)
        ENDIF

        nterms = n_elements(a)         ; # of parameters
        nfree = n_elements(y) - nterms ; Degrees of freedom
```

```
        IF nfree LE 0 THEN message, 'Curvefit - not enough data points.'

        flambda = 0.001                    ;Initial lambda
        diag = lindgen(nterms)*(nterms+1) ; Subscripts of diagonal elements

;       Define the partial derivative array

        IF double THEN pder = dblarr(n_elements(y), nterms) $
        ELSE pder = fltarr(n_elements(y), nterms)
;
        FOR iter = 1, itmax DO BEGIN      ; Iteration loop

;          Evaluate alpha and beta matricies.

           IF keyword_set(NODERIVATIVE) THEN BEGIN

;             Evaluate function and estimate partial derivatives
              CALL_PROCEDURE, Function_name, x, a, a1, yfit

              FOR term=0, nterms-1 DO BEGIN

                 p = a       ; Copy current parameters

                 ; Increment size for forward difference derivative
                 inc = eps * abs(p[term])
                 IF (inc EQ 0.) THEN inc = eps
                 p[term] = p[term] + inc
                 CALL_PROCEDURE, function_name, x, p, a1, yfit1
                 pder[0,term] = (yfit1-yfit)/inc
```

```
   ENDFOR
ENDIF ELSE BEGIN

   ; The user's procedure will return partial derivatives
   call_procedure, function_name, x, a, a1, yfit, pder

ENDELSE

IF nterms EQ 1 THEN pder = reform(pder, n_elements(y), 1)

beta = (y-yfit)*Weights # pder
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1)*pder)

; save current values of return parameters

sigma1 = sqrt( 1.0 / alpha[diag] )           ; Current sigma.
sigma  = sigma1

chisq1 = total(Weights*(y-yfit)^2)/nfree; Current chi squared.
chisq = chisq1

yfit1 = yfit
pder1 = pder

done_early = chisq1 LT total(abs(y))/1e7/NFREE
IF done_early THEN GOTO, done

c = sqrt(alpha[diag])
c = c # c
```

```
lambdaCount = 0

REPEAT BEGIN

   lambdaCount = lambdaCount + 1

   ; Normalize alpha to have unit diagonal.

   array = alpha / c

   ; Augment the diagonal.

   array[diag] = array[diag]*(1.+flambda)

   ; Invert modified curvature matrix to find new parameters.

   IF n_elements(array) EQ 1 THEN array = (1.0 / array) $
   ELSE array = invert(array)

   b = a + array/c # transpose(beta)          ; New params

   ; Evaluate function
   call_procedure, function_name, x, b, a1, yfit
   chisq = total(Weights*(y-yfit)^2)/nfree    ; New chisq
   sigma = sqrt(array[diag]/alpha[diag])      ; New sigma

   IF (finite(chisq) EQ 0) OR $
        (lambdaCount GT 30 AND chisq GE chisq1) THEN BEGIN
```

```
            ; Reject changes made this iteration, use old values.

            yfit  = yfit1
            pder = pder1
            sigma = sigma1
            chisq = chisq1

            message, 'Failed to converge', /INFORMATIONAL
            curve_err=0
            GOTO, done

         ENDIF

         flambda = flambda*10.              ; Assume fit got worse

      ENDREP UNTIL chisq LE chisq1

      flambda = flambda/100.
      a=b                           ; Save new parameter estimate.

      IF ((chisq1-chisq)/chisq1) LE tol THEN GOTO,done   ;Finished?
   ENDFOR                        ;iteration loop
;
   MESSAGE, 'Failed to converge', /INFORMATIONAL
   curve_err=0
;
done:  chi2 = chisq ; Return chi-squared (chi2 obsolete-still works)
   IF done_early THEN iter = iter - 1
   return,yfit          ; return result
END
```

fgauss3.pro

```
pro fgauss3, x, b, f, pder

del= x - b (1)
inside = -0.5* (del / b(2))^2
xpterm= exp(inside)
f=b(0)*xpterm+b(3)

pder = [ [ xpterm ],$
         [ b(0)*del*xpterm/(b(2))^2 ],$
   [ b(0)*del^2*xpterm/(b(2))^3 ], $
         [ replicate(1.0 , n_elements(x)] ]

end
```

f_emiss_wide.pro

```
pro f_emiss_wide, x, b, c, f, pder
n_c=n_elements(c)
n_el=n_elements(x)

dx=x(1)-x(0)
x1=x(0)-n_c*dx+findgen(n_el+2*n_c)*dx
n_x1=n_elements(x1)
a=fltarr(n_x1,n_x1)

del1= x1 - b (1)
inside1 = -0.5* (del1 / b(2))^2
xpterm1= exp(inside1)

f1=(b(0)/b(2))*xpterm1+b(3)

pder1 = [ [ xpterm1/b(2) ],$
          [ b(0)/b(2)*del1*xpterm1/(b(2))^2 ],$
      [ b(0)/b(2)*del1^2*xpterm1/(b(2))^3 - $
        b(0)/(b(2))^2*xpterm1   ],$
      [ replicate(1.0,n_x1)]]
pder2=pder1

;filling arrays
for i=0,n_x1-1 do $
   a((i-n_c/2)>0:(i+n_c/2)<(n_x1-1),i)=c((n_c/2-i)>0: $
                              (n_c/2+(n_x1-i-1))<(n_c-1))
a=transpose(a)
f1=transpose(a##f1)
```

```
pder1=transpose(a ## transpose(pder1))

f=f1(n_c:(n_c+n_el-1))
pder=pder1(n_c:(n_c+n_el-1),*)

end
```

f_emiss_wide_three_cV_227.pro

```
pro f_emiss_wide_three_cV_227, x, b, c, f, pder

n_c=n_elements(c)
n_el=n_elements(x)

dx=x(1)-x(0)
x1=x(0)-n_c*dx+findgen(n_el+2*n_c)*dx
n_x1=n_elements(x1)
a=fltarr(n_x1,n_x1)

del1= x1 - b(1); C V 227.089,Rel I=40
inside1 = -0.5* (del1 / b(2))^2
xpterm1= exp(inside1)

del2= x1 - b(1) -.638;C V 227.727,Rel I=5
inside2 = -0.5* (del2 / b(2))^2
xpterm2= exp(inside2)

del3= x1 - b(1) -.703; 227.792,Rel I=20
inside3 = -0.5* (del3 / b(2))^2
xpterm3= exp(inside3)

f1=b(0)/b(2)*(1.*xpterm1+.125*xpterm2+.5*xpterm3)+b(3)

pder1 = [ [ (1.*xpterm1+.125*xpterm2+.500*xpterm3)/b(2) ],$
          [ b(0)*(1.000*del1*xpterm1/(b(2))^3 + $
                  0.125*del2*xpterm2/(b(2))^3 + $
                  0.500*del3*xpterm3/(b(2))^3)  ],$
```

```
          [ -b(0)*(1.*xpterm1+.125*xpterm2+.500*xpterm3)/b(2)^2+ $
         b(0)*(1.000*del1^2*xpterm1/b(2)^4 + $
               0.125*del2^2*xpterm2/b(2)^4 + $
               0.500*del3^2*xpterm3/b(2)^4) ],$
      [ replicate(1.0,n_x1)]]

;filling arrays
for i=0,n_x1-1 do $
   a((i-n_c/2)>0:(i+n_c/2)<(n_x1-1),i)=c((n_c/2-i)>0: $
                           (n_c/2+(n_x1-i-1))<(n_c-1))
a=transpose(a)
f2=transpose(a##f1)
pder2=transpose(a ## transpose(pder1))

f=f2(n_c:(n_c+n_el-1))
pder=pder2(n_c:(n_c+n_el-1),*)
end
```

C.4.3 Analysis codes

The analysis of the IDS data was discussed in Sec. 5.7.3 *ids_temp_fin3.pro* determines which data points to attempt a fit. The plasma parameters are then found. The plasma parameters and the errors are then written to the trees.

The corrections made to the ICCD images are discussed in Sec. 5.7.4. Four codes are used to place the images in the tree. *w_iccd_data_pro.pro* has the ICCD setup parameters passed to it for a given pulse. It then calls *iccd_reduce.pro*, which applies the calibration factors to the image. The nodes are entered into the tree with *repair_iccd_node.pro* and the data is entered into the nodes with *iccd_save.pro*.

find_iccd_raw_param_SYIAH.pro determines when the chord integrated parameters should be calculated on a pulse. The criteria used are discussed in Sec. 5.7.4. The chord integrated plasma parameters are then found with *iccd_raw_multi_emis.pro*. When chord emissivity is peaked, the local plasma parameters can be found with the method described in Sec. 5.7.5. *find_iccd_shell_param_SYIAH.pro* tests the chord integrated emissivity profile, calculates the local parameters and writes them to the tree. The location of the Z-pinch is determined with *find_center_SYIAH.pro*, which calculates the local plasma parameters for various center locations and determines the best center location by comparing the emissivity and velocity of the inner two shells. The local parameters are found with *shell_par_SYIAH.pro*. The length matrix used is calculated with *shell_length_mid.pro*.

All of these run with minimal user intervention. This was done to remove any bias that the user may have during the analysis of the data. The simulated spectral intensities are stored in the trees. The profiles can be verified by comparing the measured and calculated spectral intensities. The local and chord integrated parameter profiles should also be compared. The local profiles often have steeper gradients. The general shapes of the profiles should be similar.

ids_temp_fin3.pro

```
pro ids_temp_fin3,shot=shot,temp=temp,sigtemp=sigtemp, $
    amp=amp,sigamp=sigamp, $
    lamb0=lamb0,siglamb0=siglamb0,time=time,slit=slit,ids_sm=ids_sm, $
    tim1=tim1,lambda=lambda,dial=dial,cdial=cdial, $
    lamb_ideal=lamb_ideal,sm_1=sm_1,sm_2=sm_2, $
    t_start=t_start,t_end=t_end,min_v_sig,nterms=nterms, $
    c_dial=c_dial,sigcdial=sigcdial,vel=vel,sigvel=sigvel,mass=mass, $
    view=view
;
; Default values for logbook parameters
;
default,shot,331051 ;Shot Number
default,slit,10 ;slit width (micro meters)
default,mass,15.999 ;Mass of ion (Amu)
default,dial,2783.7 ;Dial setting in (A)
default,tsigmin,1.e9 ;minimum value to use for measured fwhm
;default,siginsignu,0.011e-10 ;Unknown constant used in temp err
default,lamb_ideal,2781.01*1e-10 ;Ideal Lambda of the line(m)
default,sm_1,1 ;First smooth number for double smooth
default,sm_2,1 ;Second smooth number for double smooth
default,t_start,10 ;Time to start calculating tempurature (microseconds)
default,t_end,80 ;Time to stop calculating tempurature (microseconds)
default,min_v_sig,-.005 ;minimun signal to attempt temperature calculation
default,nterms,4 ;number of terms to use in gaussfit
default,cdial,0 ;difference between lamb0 and lamb_lideal on 90 shots (A)
default,sigcdial,0              ;error in cdial (A)
default,view,'90';viewing port
```

```
if view eq '35' then view_a=35 else view_a=0
;
;setting shot,getting time base
;
ss,shot
ids1=data('\digitizers::zap_3412_a:input_1',xaxis=tim)
nel=n_elements(ids1)
tim=tim*1e6 ;adjusting time array (micro seconds)
;
;Initialize Constants
;
nch=16 ;Number of channels

;channel names of the raw data
ch_nam=['\digitizers::zap_3412_a:input_1', $
         '\digitizers::zap_3412_a:input_2', $
         '\digitizers::zap_3412_a:input_3', $
         '\digitizers::zap_3412_a:input_4', $
         '\digitizers::zap_3412_b:input_1', $
         '\digitizers::zap_3412_b:input_2', $
         '\digitizers::zap_3412_b:input_3', $
         '\digitizers::zap_3412_b:input_4', $
         '\digitizers::zap_3412_c:input_1', $
         '\digitizers::zap_3412_c:input_2', $
         '\digitizers::zap_3412_c:input_3', $
         '\digitizers::zap_3412_c:input_4', $
         '\digitizers::zap_3412_d:input_1', $
         '\digitizers::zap_3412_d:input_2', $
         '\digitizers::zap_3412_d:input_3', $
         '\digitizers::zap_3412_d:input_4']
```

```
;scale factors (independent of all parameters)
cal_fac=[1.09723375 ,0.900595875,0.960089125,0.966937875, $
         0.966911125,0.989201375,0.981266875,1,           $
         1.012489375,1.0764375  ,1.2428     ,0.963335125, $
         0.990833375,1.09686625 ,1.040672625,1.29062375]
sigcal_fac=[0.001747631,0.001948167,0.002843711,0.003193323, $
            0.002954162,0.002672477,0.002533517,0.002270627, $
            0.00217039 ,0.002242405,0.002774807,0.002451812, $
            0.002526445,0.002753236,0.00317802 ,0.003398917]

;
;Initializing Arrays
;
ids=fltarr(nel,16)
sig_d=fltarr(nel,16)
ids_low=fltarr(nel,16)
ids_high=fltarr(nel,16)
;
;Parameters from IDS calibration
;
;average voltage signal sigma
sigsig=0.001 ;error in the voltage measurement(moise level of digitizers)

x=[0.320817838,0.297842325,0.2759533  ,0.255078138,0.234059788, $
   0.211890013,0.189719638,0.167228563,0.144396963,0.121756688, $
   0.100109688,0.07861185 ,0.05763135 ,0.037850938,0.018704063,0]
sigx=[0.001593992,0.001457234,0.001425688,0.001376474,0.001322406, $
      0.001273049,0.001237905,0.001184458,0.001149112,0.001111576, $
      0.00105742 ,0.001018223,0.000925628,0.000754417,0.000591216,0]
```

```
sigx=total(sigx)/n_elements(sigx)*1e-9   ;Error in the dispersion in (m)

;delta lambda between channel 16 and the dial (A)
dialcor=-4.64+cdial

;Calculating lambda of each channel (m)
lambda=(x+(dial+dialcor)*.1)*1e-9
siglambda=sigx*1e-9                      ;need to add errors for dial
;
;Full width half max of each channel (nm)
;
;fwhm for 10 nm and 30 nm slit widths
if (slit eq 10) then begin
  fwhm=[0.028284475,0.026712775,0.024897525,0.024623025,0.02525235, $
        0.025597475,0.02563125 ,0.026085325,0.026458425,0.02683885, $
        0.02579785 ,0.025925625,0.02580485 ,0.025507275,0.02513725, $
        0.024972225]
  sigfwhm=[0.001123507,0.001422191,0.002295178,0.002570645,0.002174236, $
           0.001674691,0.001839515,0.001437365,0.001475303,0.001679625, $
           0.001995407,0.0019059  ,0.002031325,0.002193243,0.002988289, $
           0.003014583]

  endif else if (slit eq 30) then begin
  fwhm=[0.030689267,0.028936433,0.027746267,0.0279142  ,0.028586967, $
        0.028432967,0.0282238  ,0.028575067,0.029159167,0.0295244  , $
        0.028902733,0.028369667,0.027955467,0.027846833,0.027795733, $
        0.027883567]
  sigfwhm=[0.001662433,0.001665302,0.002910932,0.003311828,0.002975127, $
           0.003124332,0.002882961,0.002641393,0.002325255,0.002400108, $
           0.003241493,0.002796515,0.00283306 ,0.003137584,0.003252324, $
```

```
           0.00371306]

  endif

fwhmI=fwhm*1e-9*3/7 ;converting to (m) A2 units
sigfwhmI=sigfwhm*1e-9*3/7 ;converting to (m) A2 units
;
;Physical Constants
;
alpha = mass * 936800890. ;gaussian constant = mc^2/e,
c = 299792458 ;speed of light in m/sec
nu_ideal=c/lamb_ideal ;ideal frequency (s)-1
nu = c / lambda ;frequencies
signu = nu * siglambda / lambda ;error in nu
FWHMnuI_b=nu*fwhmI/lambda ;FWHM machine in nu

FWHMnuI=total(FWHMnuI_b)/nch
sigFWHMnuI=sqrt((signu*fwhmI/lambda)^2+(sigfwhmI*nu/lambda)^2+ $
                (siglambda*nu*fwhmI/lambda^2)^2)
sigFWHMnuI=total(sigFWHMnuI)/sqrt(nch)
;
;Reading and smoothing data from the tree
;
for i = 0, nch-1 do begin
   ids(0:nel-1,i)=data(ch_nam(i))*cal_fac(i)

   ; Get coefficients:
   Flow=0
   Fmid=.2
   Fhigh=1
```

```
   a=50
   N_terms=13
   Coeff_low = DIGITAL_FILTER(Flow, Fmid, A, N_terms)
   Coeff_high= DIGITAL_FILTER(Fmid, Fhigh, A, N_terms)
   ; Apply the filter:
   ids_low(0:nel-1,i) = CONVOL(ids(0:nel-1,i), Coeff_low)
   ids_high(0:nel-1,i) = CONVOL(ids(0:nel-1,i), Coeff_high)
   for j=0,nel-1 do $
      sig_d(j,i)=(max(ids_high((j-10)>0:(j+10)<(nel-1),i))- $
                  min(ids_high((j-10)>0:(j+10)<(nel-1),i)))/2
endfor
m_1=data('\m_1_p0',xaxis=tim_m)
;
;test data
;
;for i=1,nch do $
;   ids_low(nel-1,i-1)=-exp(-((i-7)/4)^2)
;
;Finding data during the region of interest
;
n_start=min(where(tim gt t_start)) ;element to start using data
n_end=max(where(tim lt t_end)) ;element to stop using data
time=tim(n_start:n_end) ;adjusting time base
ids_use=ids_low(n_start:n_end,*) ;adjusting data base
sig_d_use=sig_d(n_start:n_end,*)
nuse=n_elements(time) ;finding total number of elements of reduced array
;
;Initializing fit arrays
;
par=fltarr(nuse,3) ;Array of saved fit coefficients
```

```
sig=fltarr(nuse,3) ;Array of saved fit errors
a=fltarr(3) ;Array of coefficients returned from fit
yfit=fltarr(nch) ;dummy array (fitted data)
amp=fltarr(nuse) ;Amplitude of fitted curve
lamb0=fltarr(nuse) ;Center wavelength of fitted data
nu0=fltarr(nuse)
signu0=fltarr(nuse)
FWHMnu=fltarr(nuse)
sigFWHMnu=fltarr(nuse)
tsig=fltarr(nuse)  ;Square root of delta lambda minus machine width
temp=fltarr(nuse)   ;Temperature (eV)
temp1=fltarr(nuse) ;Temperature (eV)
ids_use_min=fltarr(nuse) ;Minimum voltage at each time
cal_ch=fltarr(nuse) ;Cannel of the minimun voltage
errc=intarr(nuse)
;
;Finding data with a large enough signal that curve can calculate a signal
;
for i = 0,nuse-1 do begin
   ids_use_min(i)=min(ids_use(i,3:12))
   cal_ch(i)=where(ids_use_min(i) eq min(ids_use(i,3:12)))+3
endfor

cal=where(ids_use_min lt min_v_sig)
n_cal=n_elements(cal)
if cal(0) eq -1 then cal(0)=1
;
;curve fit for each time of interest
;
for i=1,n_cal-1 do begin
```

```
   y=ids_use(cal(i),*)
   sig_y=sig_d_use(cal(i),*)
   yfit=gaussfit(x,y,b,nterms=nterms)  ;Intitial guess using guassfit
   if ((b(0) gt 0) or (b(0) lt -1.5)) then begin
       b(0:2)=a(0:2)
       b(3)=y(0)>y(15)
       endif else $
       b(3)=y(0)>y(15)>b(3)

   a=b(0:2)
   ;weights for curve fit 1/sigma^2
   w=1./ ( (cal_fac*(sig_y>.001))^2 +(y*sigcal_fac)^2)
   y=y-b(3)
   yfit=err_curvefit(x,y,w,a,sigma,function_name='fgauss3', $
                     curve_err=curve_err);,tol=1e-4)
   par(cal(i),*)=a
   sig(cal(i),*)=sigma
   errc(cal(i))=curve_err
endfor

amp=par(*,0) ;Amp(V)
sigamp=sig(*,0) ;Error Amp (V)
cal1=where(amp lt -sigamp)

lamb0=par(*,1)*1e-9 + lambda(15) ;Centroid of the gaussian (m)
siglamb0=sig(*,1)*1e-9 ;Error of the centroid (m)
if cal1(0) ne -1 then begin
   cal_temp=(where((lamb0(cal1) - siglamb0(cal1)) gt lambda(15)))
   if cal_temp(0) eq -1 then $
   cal2=[0] else $
```

```
   cal2=cal_temp
   endif else cal2=[-1]
if cal2(0) ne -1 then $
   cal4=cal2(where((lamb0(cal2) + siglamb0(cal2)) lt lambda(0))) $
   else cal4=[0]

cal_temp=where(errc(cal4) eq 1)
if cal_temp(0) eq -1 then $
   cal3=[0] else $
   cal3=cal4(cal_temp)
time=time(cal3)

lamb0=lamb0(cal3) ;Centroid of the gaussian (m)
siglamb0=siglamb0(cal3) ;Error of the centroid (m)

amp=amp(cal3) ;Amp(V)
sigamp=sigamp(cal3) ;Error Amp (V)

FWHM=par(cal3,2)*1e-9 ;FWHM (m)
sigFWHM=sig(cal3,2)*1e-9 ;Error in FWHM (m)

nu0=c/lamb0
signu0=nu*siglamb0/lamb0

FWHMnu=FWHM*nu0/lamb0
sigFWHMnu=sigFWHM*nu0/lamb0

tsig = sqrt(FWHMnu^2-FWHMnuI^2);>tsigmin
temp = alpha*(tsig/nu_ideal)^2
```

```
;An expansion which is close enough for this
;sigtsig = FWHMnu*sigFWHMnu*$
; ( 1. + (signu*siginsignu/FWHMnu/sigFWHMnu)^2 ) / tsig
;
;sigtemp=2*alpha*( tsig / nu0^2 ) * $
;    sqrt( sigtsig^2 + (tsig*signu0/nu0 )^2 )
;
tempsig = alpha*( FWHMnuI / ( nu_ideal ) )^2

dTdFWHM=2*alpha*FWHMnu/nu_ideal^2
dTdTi=2*alpha*FWHMnuI/nu_ideal^2
sigTemp=sqrt((dTdFWHM*sigFWHMnu)^2+(dTdTi*sigFWHMnuI)^2)

vel=(lamb_ideal-lamb0)*c/(lamb_ideal*cos(view_a*!pi/180))
sigvel=sqrt(siglamb0^2+sigx^2+(sigcdial*1e-10)^2)*c/ $
       (lamb_ideal*cos(view_a*!pi/180))

window,1,xsize=600,ysize=250,xpos=0,ypos=600,title='Window 1'
window,2,xsize=600,ysize=250,xpos=0,ypos=350,title='Window 2'
window,3,xsize=600,ysize=250,xpos=0,ypos=100,title='Window 3'
window,4,xsize=600,ysize=250,xpos=600,ypos=100,title='Window 4'
window,5,xsize=600,ysize=500,xpos=600,ypos=400,title='Window 5'
wset,1
plot,time ,amp ,$
           title='Amplitude (V)',xtitle='Time', $
           /yst,/xst,xra=[t_start,t_end], $
           yra=[-.5,0]
errplot,time ,amp -sigamp ,amp +sigamp

wset,2
```

```
plot,time ,lamb0 ,yra=[min(lambda),max(lambda)] , $
          title='Peak Wavelength',xtitle='Time', $
          /yst,/xst,xra=[t_start,t_end]
errplot,time ,lamb0 -siglamb0 ,lamb0 +siglamb0
hline,lamb_ideal,color=2

wset,3
plot,time ,temp ,$
           title='Temperatue (eV)  Shot='+stremo(shot), $
           xtitle='Time',/yst,/xst,$
       xra=[t_start,t_end],yra=[0,1000]
errplot,time ,temp -sigtemp ,temp +sigtemp
hline,tempsig,color=2

wset,4
plot,time ,vel ,$
           title='Velocity (m/s)  Shot='+stremo(shot), $
           xtitle='Time',/yst,/xst,$
       xra=[t_start,t_end],yrange=[0,2e5]
errplot,time ,vel -sigvel ,vel +sigvel

wset,5
plot,time ,vel ,$
           title='Velocity (m/s)  Shot='+stremo(shot), $
           xtitle='Time',/yst,/xst,$
       xra=[t_start,t_end],yrange=[-10000,10000]
hline,0
if n_elements(tim_m) ne 0 then oplot,tim_m*1e6,m_1*1e5,color=2

;
```

```
;Routine used to find CDial
;
;c_dial=[(-total(lamb0(where((time gt 25) and (time lt 30))))/ $
;       (n_elements(where((time gt 25) and (time lt 30)))) $
;       +lamb_ideal)*1e10 + cdial, $;
;;
;       (-total(lamb0(where((time gt 30) and (time lt 35))))/ $
;       (n_elements(where((time gt 30) and (time lt 35)))) $
;       +lamb_ideal)*1e10 + cdial, $
;;
;        (-total(lamb0(where((time gt 35) and (time lt 40))))/ $
;       (n_elements(where((time gt 35) and (time lt 40)))) $
;       +lamb_ideal)*1e10 + cdial, $
;;
;        (-total(lamb0(where((time gt 40) and (time lt 45))))/ $
;       (n_elements(where((time gt 40) and (time lt 45)))) $
;       +lamb_ideal)*1e10 + cdial]
;;
;print,'shot= ',shot,'(25,30,35,40)  cdial=',c_dial
tim1=time*1e-6

end
```

w_iccd_data_pro.pro

```
pro w_iccd_data,shot=shot,port=port,grate=grate,gate=gate, $
                gain=gain,trans_focus=trans_focus,trans_r=trans_r, $
                wave_set=wave_set,comment=comment

ss,shot
port_i=-1
grating_selection=-1

if port eq "side" then port_i = 1
if port eq "front" then port_i = 0

if grate eq 150 then grat_i = 2
if grate eq 2400 then grat_i = 1
if grate eq 3600 then grat_i = 0

print,'reducing'

iccd_reduce,shot=shot,port_i=port_i,grat_i=grat_i, $
                image=image,imag_c=image_c,pix=pix, $
                intens=intens,inten_cor=inten_cor,wave=wave, $
                row_offset=row_offset,bin_min=bin_min,bin_max=bin_max, $
                rel_i=rel_i,fwhm_i=fwhm_i,inst_line=inst_line

print,'Repairing tree'

repair_iccd_node,shot

print,'saving'
```

```
iccd_save,shot=shot,port_i=port_i,gate=gate, grating=grate, gain=gain, $
              wavs=wave_set, pix=pix, $
              intens=intens, iccd_image=image, wave=wave, $
              row_offset=row_offset,bin_min=bin_min,bin_max=bin_max, $
              rel_i=rel_i,fwhm_i=fwhm_i,inst_line=inst_line, $
              comment=comment

print,'finding velocities'
find_iccd_raw_param_SYIAH,shot
find_iccd_shell_param_SYIAH,shot

end
```

iccd_reduce.pro

```
pro iccd_reduce,shot=shot,port_i=port_i,grat_i=grat_i, $
                image=image,imag_c=imag_c,pix=pix,$
                intens=intens,inten_cor=inten_cor,wave=wave, $
                row_offset=row_offset,bin_min=bin_min,bin_max=bin_max, $
                rel_i=rel_i,fwhm_i=fwhm_i,inst_line=inst_line



default,dat_tex,'26jul00'
default,shot_num,25
default,shot,726025
default,port_i,1
default,grat_i,0

case !version.os_family of
  'unix':    sep_char = '/'
  'Windows': sep_char = '\'
  else:      begin
          print, 'ICCD not supported on '+!version.os_family
       return
       end
  endcase

;reading data

default, iccd_filepath, getenv('iccd_filepath')

if iccd_filepath eq '' then begin
```

```
  print, 'Please set your iccd_filepath enviroment variable, e.g.'
  print, "IDL> setenv,'iccd_filepath=j:\Zap Spec Data\iccd_data'"
  print, 'where j: is your disk share to zappc1'
  return
  endif

if shot eq 0 then $
  shotname = stremo( mdsvalue('$shotname', /quiet, status=status) ) $
else $
  shotname = stremo(shot)

; convert shotname to ICCD SPE data structure format

sn_len  = strlen( shotname )

shot_no = strmid( shotname, 2, 3, /reverse )
day     = strmid( shotname, 4, 2, /reverse )

case sn_len of
  6:  begin
        month = '0' + strmid( shotname, 6, 1, /reverse )
year  = '00'
end
  7:  begin
        year  = '00'
        month = strmid( shotname, 6, 2, /reverse )
end
  8:  begin
        year  = '0' + strmid( shotname, 7, 1, /reverse )
        month = strmid( shotname, 6, 2, /reverse )
```

```
end
  9:  begin
        year  = strmid( shotname, 7, 2, /reverse )
        month = strmid( shotname, 6, 2, /reverse )
end
  endcase

month_name = [  'jan', 'feb', 'mar', 'apr', 'may', 'jun', $
'jul', 'aug', 'sep', 'oct', 'nov', 'dec' ]

if year eq '03' then begin
  year_folder_name = ''
endif else if year eq '02' then begin
  year_folder_name = '2002\'
endif else if year eq '01' then begin
  year_folder_name = '2001\'
endif else begin
  year_folder_name = '2000\'
endelse

if year eq '03' then begin
  folder_name = year + month_name( fix(month-1) ) + day
endif else begin
  folder_name = day + month_name( fix(month-1) ) + year
endelse

if strmid( shot_no, 0, 1 ) eq '0' then $
  shot_no = strmid( shot_no, 1 )

if strmid( shot_no, 0, 1 ) eq '0' then $
```

```
  shot_no = strmid( shot_no, 1 )

if strmid( iccd_filepath, 0, 1, /reverse ) ne sep_char then $
  iccd_filepath = iccd_filepath + sep_char

filename=iccd_filepath+year_folder_name+folder_name+sep_char+ $
         folder_name+shot_no+'.SPE'

read_princeton, filename, image, $
x_calibration=wave

pix=findgen(n_elements(wave))
;correction arrays
n_el=512 ;number of pixels
imag_c=fltarr(n_el,n_el) ;array of row corrected data

ymd=fix(day)+100*fix(month)+10000*fix(year)

if ymd lt 010823 then begin
  restore,iccd_filepath+'iccd_cal_param.dat'
  ;chosing correct row corrections

  if wave(n_el/2) le 275 then wav_i=0 $
     else if wave(n_el/2) le 340 then wav_i=1 $
     else if wave(n_el/2) le 420 then wav_i=2 $
     else if wave(n_el/2) le 520 then wav_i=3 $
     else if wave(n_el/2) le 600 then wav_i=4 $
     else if wave(n_el/2) le 675 then wav_i=5 $
     else wav_i=6
endif else begin
```

```
  if ymd lt 021007 then begin
    restore,iccd_filepath+'iccd_cal_param_18sep01.dat'
    ;chosing correct row corrections

    if wave(n_el/2) le 275 then wav_i=0 $
       else if wave(n_el/2) le 340 then wav_i=1 $
       else if wave(n_el/2) le 420 then wav_i=2 $
       else if wave(n_el/2) le 495 then wav_i=3 $
       else if wave(n_el/2) le 575 then wav_i=4 $
       else if wave(n_el/2) le 675 then wav_i=5 $
       else wav_i=6
    print,wav_i,port_i,grat_i,0
  endif else begin
    restore,iccd_filepath+'iccd_cal_param_ri_17oct02.dat'
    ;chosing correct row corrections

    if wave(n_el/2) le 275 then wav_i=0 $
       else if wave(n_el/2) le 340 then wav_i=1 $
       else if wave(n_el/2) le 420 then wav_i=2 $
       else if wave(n_el/2) le 495 then wav_i=3 $
       else if wave(n_el/2) le 575 then wav_i=4 $
       else if wave(n_el/2) le 675 then wav_i=5 $
       else wav_i=6
    added 16 Jan 03 since the calibrations for 5,6 are bad
    if (wav_i ge 4) and (grat_i eq 1) then wav_i=4;
    print,wav_i,port_i,grat_i,1
  endelse
endelse

row_offset=row_offset(*,wav_i,port_i,grat_i)
```

```
bin_min=bin_min(*,wav_i,port_i,grat_i)
bin_max=bin_max(*,wav_i,port_i,grat_i)
rel_i=rel_i(*,wav_i,port_i,grat_i)
fwhm_i=fwhm_inst(*,wav_i,port_i,grat_i)
inst_line=inst_func(*,*,wav_i,port_i,grat_i)

if max(rel_i) eq 0 then rel_i=rel_i+1.0

for j=0,n_el-1 do begin
     imag_c(*,j)=interpol(transpose(image(*,j)),pix-row_offset(j),pix)
endfor

intens=fltarr(n_el,20)
inten_cor=fltarr(n_el,20)

for j=0,19 do begin
  for i=0,n_el-1 do begin
    intens(i,j)=total(imag_c(i,bin_min(j):bin_max(j)))
  endfor
  inten_cor(*,j)=intens(*,j)*rel_i(j)
endfor

end
```

repair_iccd_node.pro

```
pro repair_iccd_node,shot
if shot le 21126025 then begin
  mdstcl,"edit zapmain/shot="+stremo(shot)
  mdstcl,"set def .signals"
  mdstcl,"delete node iccd /noconfirm"
  mdstcl,"add node/usage=structure .iccd"
;
  mdstcl,"set def \zapmain::top.signals.iccd"
  mdstcl,"add node/usage=signal  iccd_mon"
  mdstcl,"add tag iccd_mon iccd_mon"

  mdstcl,"write"
  mdstcl,"close"
endif

mdstcl,"edit zapanalysis/shot="+stremo(shot)
mdstcl,"add node/usage=structure .iccd"
mdstcl,"set def .iccd"
mdstcl,"add node/usage=structure .spectra"
mdstcl,"set def .spectra"
mdstcl,"add node/usage=text comment"
mdstcl,"add node/usage=text enter_port"
mdstcl,"add node/usage=numeric gain"
mdstcl,"add node/usage=numeric gate"
mdstcl,"add node/usage=numeric grating"
mdstcl,"add node/usage=signal  iccd_image"
mdstcl,"add node/usage=text    view"
mdstcl,"add node/usage=numeric wavelength"
```

```
mdstcl,"add node/usage=structure .bin_param"
mdstcl,"add node/usage=structure .tele_35"
mdstcl,"add node/usage=structure .tele_90"
mdstcl,"add node/usage=structure .intensity"
mdstcl,"set def \zapanalysis::top.iccd.spectra.bin_param"
mdstcl,"add node/usage=signal row_offsets"
mdstcl,"add node/usage=signal start_bin"
mdstcl,"add node/usage=signal end_bin"

mdstcl,"set def \zapanalysis::top.iccd.spectra.tele_35"
mdstcl,"add node/usage=numeric trans_focus"
mdstcl,"add node/usage=numeric trans_r"
mdstcl,"add node/usage=numeric iris_diam"
mdstcl,"add node/usage=text iris_hole_se"

mdstcl,"set def \zapanalysis::top.iccd.spectra.tele_90"
mdstcl,"add node/usage=numeric trans_focus"
mdstcl,"add node/usage=numeric trans_r"
mdstcl,"add node/usage=numeric iris_diam"
mdstcl,"add node/usage=text iris_hole_se"

mdstcl,"set def \zapanalysis::top.iccd.spectra.intensity"
for i=1,9 do begin
   mdstcl,"add node/usage=signal iccd_0"+stremo(i)
   mdstcl,"add tag iccd_0"+stremo(i)+" iccd_0"+stremo(i)
   mdstcl,"add node/usage=signal iccd_0"+stremo(i)+":raw_binned"
   mdstcl,"add node/usage=numeric iccd_0"+stremo(i)+":scale_fact"
   mdstcl,"add node/usage=signal iccd_0"+stremo(i)+":inst_func"
   mdstcl,"add node/usage=numeric iccd_0"+stremo(i)+":inst_fwhm"
   mdstcl,'put iccd_0'+stremo(i)+' "\ICCD_0'+stremo(i)+ $
```

```
          ':RAW_BINNED * \ICCD_0'+stremo(i)+':SCALE_FACT"'
endfor
;stop
for i=10,20 do begin
   mdstcl,"add node/usage=signal iccd_"+stremo(i)
   mdstcl,"add tag iccd_"+stremo(i)+" iccd_"+stremo(i)
   mdstcl,"add node/usage=signal iccd_"+stremo(i)+ $
          ":raw_binned"
   mdstcl,"add node/usage=numeric iccd_"+stremo(i)+ $
          ":scale_fact"
   mdstcl,"add node/usage=signal iccd_"+stremo(i)+ $
          ":inst_func"
   mdstcl,"add node/usage=numeric iccd_"+stremo(i)+ $
          ":inst_fwhm"
   mdstcl,'put iccd_'+stremo(i)+' "\ICCD_'+stremo(i)+ $
          ':RAW_BINNED * \ICCD_'+stremo(i)+':SCALE_FACT"'
endfor
mdstcl,"add node/usage=signal iccd_lambda"
mdstcl,"add tag iccd_lambda iccd_lambda"
mdstcl,"add tag \zapanalysis::top.iccd iccd"

mdstcl,"write"
mdstcl,"close"

ss,shot
mdsput, '.zapanalysis.iccd.spectra.tele_35:iris_diam', $
          'build_with_units(31.8,"")'
mdsput, '.zapanalysis.iccd.spectra.tele_35:iris_hole_se', $
          'build_with_units("D","")'
mdsput, '.zapanalysis.iccd.spectra.tele_35:trans_focus', $
```

```
          'build_with_units(1.25,"")'
mdsput, '.zapanalysis.iccd.spectra.tele_90:iris_diam', $
          'build_with_units(22.2,"")'
mdsput, '.zapanalysis.iccd.spectra.tele_90:iris_hole_se', $
          'build_with_units("A","")'
mdsput, '.zapanalysis.iccd.spectra.tele_90:trans_focus', $
          'build_with_units(1.771,"")'
if shot lt 001031000 then begin
  mdsput, '.zapanalysis.iccd.spectra.tele_35:trans_r', $
          'build_with_units(1.90,"")'
  mdsput, '.zapanalysis.iccd.spectra.tele_90:trans_r', $
          'build_with_units(1.94,"")'
endif else begin
  mdsput, '.zapanalysis.iccd.spectra.tele_35:trans_r', $
          'build_with_units(1.64,"")'
  mdsput, '.zapanalysis.iccd.spectra.tele_90:trans_r', $
          'build_with_units(1.68,"")'
endelse

mdsclose
end
```

iccd_save.pro

```
pro iccd_save,shot=shot,port_i=port_i,gate=gate, $
              grating=grating, gain=gain, $
              wavs=wavs, pix=pix, $
              intens=intens, iccd_image=iccd_image, wave=wave, $
              row_offset=row_offset,bin_min=bin_min,bin_max=bin_max, $
              rel_i=rel_i,fwhm_i=fwhm_i,inst_line=inst_line, $
              comment=comment
default,comment,""

if port_i eq 1 then begin
   port='side'
   view=35
   endif else if port_i eq 0 then begin
   port='front'
   view=90
   endif

n_w=n_elements(wave)
pix_i=findgen(n_elements(inst_line(*,0)))+1
build_string_1 = 'build_signal(build_with_units($1,"counts"),' + $
'*,' + $
'\iccd_lambda,,)'
build_string_2a = 'build_signal(build_with_units($1,"nm"),' + $
',' + $
',,)'
build_string_2b = 'build_signal(build_with_units($1,"pixels"),' + $
',' + $
',,)'
```

```
build_string_3 = 'build_signal(build_with_units($1,"counts"),' + $
'*,' + $
'build_with_units($2,"pixel"),,)'
build_array = 'build_signal(build_with_units($1,"counts"),' + $
'*,' + $
'\iccd_lambda' + $
'build_dim(build_window(0,' + stremo(n_w-1) + $
',' + stremo(pix(0)) + '),' + $
'build_slope(' + stremo(pix(1)-pix(0)) + ',,)))]'

print,port

ss,shot
mdsput, '.zapanalysis.iccd.spectra.enter_port', $
        'build_with_units($,"")',port
mdsput, '.zapanalysis.iccd.spectra.gain', $
        'build_with_units($,"")',gain
mdsput, '.zapanalysis.iccd.spectra.gate', $
        'build_with_units($,"s")',gate
mdsput, '.zapanalysis.iccd.spectra.grating', $
        'build_with_units($,"")',grating
mdsput, '.zapanalysis.iccd.spectra.view', $
        'build_with_units($,"")',view
mdsput, '.zapanalysis.iccd.spectra.wavelength', $
        'build_with_units($,"nm")',wavs
mdsput, '.zapanalysis.iccd.spectra.comment', $
        'build_with_units($,"")',comment
mdsput, '.zapanalysis.iccd.spectra.bin_param:row_offsets', $
        build_string_2b,row_offset
mdsput, '.zapanalysis.iccd.spectra.bin_param:start_bin', $
```

```
        build_string_2b,bin_min
mdsput, '.zapanalysis.iccd.spectra.bin_param:end_bin', $
        build_string_2b,bin_max
mdsput, '\iccd_lambda', build_string_2a, wave
for i=0,8 do begin
   mdsput, '\iccd_0'+stremo(i+1)+':raw_binned', $
           build_string_1, intens(*,i)
   mdsput, '\iccd_0'+stremo(i+1)+':inst_func', $
           build_string_3, inst_line(*,i), pix_i
   mdsput, '\iccd_0'+stremo(i+1)+':inst_FWHM', $
           'build_with_units($,"pix")',fwhm_i(i)
   mdsput, '\iccd_0'+stremo(i+1)+':scale_fact', $
           'build_with_units($,"")',rel_i(i)
endfor
for i=9,19 do begin
   mdsput, '\iccd_'+stremo(i+1)+':raw_binned', $
           build_string_1, intens(*,i)
   mdsput, '\iccd_'+stremo(i+1)+':inst_func', $
           build_string_3, inst_line(*,i), pix_i
   mdsput, '\iccd_'+stremo(i+1)+':inst_FWHM', $
           'build_with_units($,"pix")',fwhm_i(i)
   mdsput, '\iccd_'+stremo(i+1)+':scale_fact', $
           'build_with_units($,"")',rel_i(i)
endfor

mdsclose
mdsclose

end
```

find_iccd_raw_param_SYIAH.pro

```
pro find_iccd_raw_param_SYIAH,shot
default,shot,21121022
ss,shot

;getting data
chord=findgen(20)+1
build_string_1 = 'build_signal(build_with_units($1,$2),' + $
'*,' + $
'$3,,)'

if not(data_there('\iccd_01')) then goto,blank
view=mdsvalue('.zapanalysis.iccd.spectra:view')
intense=data('\iccd_01',xaxis=wave)
intense=data('\iccd_01:inst_func',xaxis=pix)
n_el=n_elements(wave)
n_inst=n_elements(pix)
intense=fltarr(n_el,20)
inst_line=fltarr(n_inst,20)
inst_fwhm=fltarr(20)
for i=0,8 do begin
   intense(*,i)=data('\iccd_0'+stremo(i+1))>(-1.e3)
   inst_line(*,i)=data('\iccd_0'+stremo(i+1)+':inst_func')
   inst_fwhm(i)=data('\iccd_0'+stremo(i+1)+':inst_fwhm')
endfor
for i=9,19 do begin
   intense(*,i)=data('\iccd_'+stremo(i+1))>(-1.e3)
   inst_line(*,i)=data('\iccd_'+stremo(i+1)+':inst_func')
   inst_fwhm(i)=data('\iccd_'+stremo(i+1)+':inst_fwhm')
```

```
endfor
fwhm_pix = inst_fwhm
wave_s=wave
intense_s=intense
inst_line_s=inst_line
inst_fwhm_s=inst_fwhm

l_edge_ave=fltarr(20)
r_edge_ave=fltarr(20)
l_edge_sigma=fltarr(20)
r_edge_sigma=fltarr(20)
for i=0,19 do begin
   l_edge_ave(i)=total(intense(1:10,i))/10
   r_edge_ave(i)=total(intense(500:509,i))/10
   l_edge_sigma(i)=sqrt(total((intense(1:10,i)-l_edge_ave(i))^2)/9)
   r_edge_sigma(i)=sqrt(total((intense(500:509,i)-r_edge_ave(i))^2)/9)
endfor

d_raw=((l_edge_sigma/(sqrt(l_edge_ave>1.)))< $
       (r_edge_sigma/(sqrt(r_edge_ave>1.))))>1.
ave_edge=(l_edge_ave+r_edge_ave)/2
inten_back=intense
for i=0,19 do inten_back(*,i)=intense(*,i)-ave_edge(i)

window,0
!p.multi=[0,1,2]
plot,wave,intense(*,0),yra=[-.01,.1]*max(intense)
for i=0,19 do oplot,wave,intense(*,i),co=2+i
hline,5*max(d_raw)
plot,wave,inten_back(*,0),yra=[-.01,.1]*max(inten_back)
```

```
for i=0,19 do oplot,wave,inten_back(*,i),co=2+i
hline,5*max(d_raw)

; looking for lines
; C
all_ions=['c3','c3','c4','c4','c5','c6','c6', $
          'n3','n3','n4','n4','n4','n5','n6','n6', $
          'o3','o3','o4','o5','o5',$
          'o6','o6','o7', $
          'He2', $
          'b3']
all_mass=[12.107  ,12.107  ,12.107  ,12.107  ,12.107  ,12.107  ,12.107  , $
          14.00674,14.00674,14.00674,14.00674,14.00674,14.00674,14.00674, $
          14.00674,15.9994 ,15.9994 ,15.9994 ,15.9994 ,15.9994 , $
          15.9994 ,15.9994 ,15.9994 , $
          4.002602, $
          10.811 ]
all_ideal_wav=[229.687 ,464.7418,465.83 ,580.131,227.089,343.369,529.1 , $
               268.218 ,409.736 ,638.075,264.562,347.872,298.078,205.61, $
               289.64  ,326.0857,375.4696,306.343,559.789,278.101, $
               381.135 ,383.424 ,255.9, $
               468.57  , $
               206.5776]
Number_lines=[1,-1,-1, 2, 3, 1, 1, $
              1, 2, 1, 3, 3, 3, 1, 1, $
              3, 3, 2, 1, 3, $
              1, 1, 1, $
              1, $
              2]  ; use -1 for the c III c IV region
f_name=['','three_cIII_465','','two_cIV_580','three_cV_227' ,'','', $
```

```
        '','two_nIII_409','','three_nIV_264','three_nIV_347', $
        'three_nV_298','','','three_oIII_326','three_oIII_375', $
        'two_oIV_306','','three_oV_278' , $
        '','','','','two_bIII_206']

n_ions=n_elements(all_ions)
n_start=intarr(n_ions)

for i=0,n_ions-1 do n_start(i)=min(where(wave ge all_ideal_wav(i)))

regions=where(n_start ge 5)
if regions(0) lt 0 then goto,blank
n_regions=n_elements(regions)
use_region=intarr(n_regions)
use_region1=intarr(n_regions)

for i=0,n_regions-1 do $
   if max(inten_back((n_start(regions(i))-10)>5: $
         (n_start(regions(i))+10)<506,*)) ge $
          5*max(d_raw) then $
     use_region1(i)=1

for i=0,n_regions-1 do $
   if total(inten_back((n_start(regions(i))-10)>5: $
            (n_start(regions(i))+10)<506,*)) ge $
     10*(sqrt(total(abs(inten_back((n_start(regions(i))-10)>5: $
         (n_start(regions(i))+10)<506,*)))>0.)* $
      ((total(d_raw)/20.)>5))>100 then $
       use_region(i)=1
```

```
print,use_region,use_region1

n_usefull=where(use_region eq 1)
if n_usefull(0) lt 0 then goto,blank
regions=regions(n_usefull)
n_regions=n_elements(regions)

for i=0,n_regions-1 do begin
   case 1 of
     Number_lines(regions(i)) eq 1 :begin
       print,'Calculating '+all_ions(regions(i))+' '+ $
             stremo(all_ideal_wav(regions(i)))+' parameters'
       iccd_raw_emis,intense,d_raw,wave,inst_line,fwhm_inst, $
                     lamb_ideal=all_ideal_wav(regions(i)), $
                     mass=all_mass(regions(i)), $
                     par_raw=par_raw,sig_raw=sig_raw,chi_raw=chi_raw, $
                     amp_raw=amp_raw,sig_amp=sig_amp_raw, $
                     vel_raw=vel_raw,sig_vel=sig_vel_raw, $
                     temp_raw=temp_raw,sig_temp=sig_temp_raw, $
                     offset_raw=offset_raw,sig_offset=sig_offset_raw, $
                     wave_fit=wave_fit,yfit_raw=yfit_raw, $
                     error_raw=error_raw
       write_raw_param,shot,all_ions(regions(i)), $
                       all_ideal_wav(regions(i)), $
                       amp_raw,vel_raw,temp_raw,offset_raw, $
                       sig_amp_raw,sig_vel_raw, $
                       sig_temp_raw,sig_offset_raw, $
                       chi_raw,error_raw,wave_fit,yfit_raw
       end
     Number_lines(regions(i)) gt 1 :begin
```

```
  print,'Calculating '+all_ions(regions(i))+' '+ $
        stremo(all_ideal_wav(regions(i)))+' parameters'
  iccd_raw_multi_emis,intense,d_raw,wave,inst_line, $
          fwhm_inst,func_name=f_name(regions(i)), $
          lamb_ideal=all_ideal_wav(regions(i)), $
          mass=all_mass(regions(i)), $
          par_raw=par_raw,sig_raw=sig_raw,chi_raw=chi_raw, $
          amp_raw=amp_raw,sig_amp_raw=sig_amp_raw, $
          vel_raw=vel_raw,sig_vel=sig_vel_raw, $
          temp_raw=temp_raw,sig_temp=sig_temp_raw, $
          offset_raw=offset_raw, $
          sig_offset=sig_offset_raw,ps=ps, $
          wave_fit=wave_fit,yfit_raw=yfit_raw, $
          error_raw=error_raw
  write_raw_param,shot,all_ions(regions(i)), $
                  all_ideal_wav(regions(i)), $
                  amp_raw,vel_raw,temp_raw,offset_raw, $
                  sig_amp_raw,sig_vel_raw, $
                  sig_temp_raw,sig_offset_raw, $
                  chi_raw,error_raw,wave_fit,yfit_raw
  end
Number_lines(regions(i)) eq -1 :begin
  if all_ions(regions(i)) eq 'c3' then begin
    if all_ideal_wav(regions((i+1)<(n_regions-1))) eq $
                  465.83 then begin
      print,'Calculating c3_c4_465 parameters'
      iccd_raw_emis_four_CIII,intense,d_raw,wave, $
                inst_line,fwhm_inst, $
                lamb_ideal_3=464.7418,lamb_ideal_4=465.830, $
                mass=mass,itmax=itmax,tol=tol, $
```

```
          par_raw=par_raw_c3_465, $
          sig_raw=sig_raw_c3_465,chi_raw=chi_raw, $
          amp_3_raw=amp_raw_c3_465, $
          sig_amp_raw_3=sig_amp_raw_c3_465, $
          vel_raw_3=vel_raw_c3_465, $
          sig_vel_raw_3=sig_vel_raw_c3_465, $
          temp_raw_3=temp_raw_c3_465, $
          sig_temp_raw_3=sig_temp_raw_c3_465, $
          amp_4_raw=amp_raw_c4_465, $
          sig_amp_raw_4=sig_amp_raw_c4_465, $
          vel_raw_4=vel_raw_c4_465, $
          sig_vel_raw_4=sig_vel_raw_c4_465, $
          temp_raw_4=temp_raw_c4_465, $
          sig_temp_raw_4=sig_temp_raw_c4_465, $
          offset_raw=offset_raw_465, $
          sig_offset=sig_offset_raw_465,ps=ps, $
          wave_raw=wave_fit,yfit_raw=yfit_raw, $
          error_raw=error
n_bad=where((error ne 1) or $
             (finite(amp_raw_c3_465) ne 1) or $
             (finite(vel_raw_c3_465) ne 1) or $
             (finite(temp_raw_c3_465) ne 1) or $
             (finite(amp_raw_c4_465) ne 1) or $
             (finite(vel_raw_c4_465) ne 1) or $
             (finite(temp_raw_c4_465) ne 1) or $
             (finite(offset_raw_465) ne 1) or $
             (amp_raw_c3_465 lt 0) or $
             (amp_raw_c4_465 lt 0))
if n_elements(n_bad) ne 20 then begin
  if n_bad(0) ne -1 then begin
```

```
  amp_raw_c3_465(n_bad)=-1
  vel_raw_c3_465(n_bad)=-1
  temp_raw_c3_465(n_bad)=-1
  amp_raw_c4_465(n_bad)=-1
  vel_raw_c4_465(n_bad)=-1
  temp_raw_c4_465(n_bad)=-1
  offset_raw_465(n_bad)=-1
  chi_raw(n_bad)=-1
endif
mdstcl,"edit zapanalysis/shot="+stremo(shot)
mdstcl,"set def .iccd"
mdstcl,"add node/usage=numeric c3"
mdstcl,"add tag c3 c3"
mdstcl,"set def c3"
mdstcl,"add node/usage=signal emiss_raw"
mdstcl,"add node/usage=signal vel_raw"
mdstcl,"add node/usage=signal temp_raw"
mdstcl,"add node/usage=signal offset_raw"
mdstcl,"add node/usage=signal emiss_raw:error"
mdstcl,"add node/usage=signal vel_raw:error"
mdstcl,"add node/usage=signal temp_raw:error"
mdstcl,"add node/usage=signal offset_raw:error"
mdstcl,"add node/usage=signal chisq_raw"
mdstcl,"add node/usage=text comment"
for i=1,9 do $
   mdstcl,"add node/usage=signal fit_raw_0"+stremo(i)
for i=10,20 do $
   mdstcl,"add node/usage=signal fit_raw_"+stremo(i)

mdstcl,"set def .iccd"
```

```
mdstcl,"add node/usage=numeric c4"
mdstcl,"add tag c4 c4"
mdstcl,"set def c4"
mdstcl,"add node/usage=signal emiss_raw"
mdstcl,"add node/usage=signal vel_raw"
mdstcl,"add node/usage=signal temp_raw"
mdstcl,"add node/usage=signal emiss_raw:error"
mdstcl,"add node/usage=signal vel_raw:error"
mdstcl,"add node/usage=signal temp_raw:error"
mdstcl,"add node/usage=text comment"
for i=1,9 do $
   mdstcl,"add node/usage=signal fit_raw_0"+stremo(i)
for i=10,20 do $
   mdstcl,"add node/usage=signal fit_raw_"+stremo(i)

mdstcl,"write"
mdstcl,"close"
mdstcl,"close"
ss,shot
mdsput, '\c3','build_with_units(464.7418,"nm")'
mdsput, '\c3:emiss_raw', build_string_1, $
        amp_raw_c3_465,"counts",chord
mdsput, '\c3:vel_raw', build_string_1, $
        vel_raw_c3_465,"m/s",chord
mdsput, '\c3:temp_raw', build_string_1, $
        temp_raw_c3_465,"eV",chord
mdsput, '\c3:offset_raw', build_string_1, $
        offset_raw_465,"counts",chord
mdsput, '\c3:emiss_raw:error', build_string_1, $
        sig_amp_raw_c3_465,"counts",chord
```

```
mdsput, '\c3:vel_raw:error', build_string_1, $
        sig_vel_raw_c3_465,"m/s",chord
mdsput, '\c3:temp_raw:error', build_string_1, $
        sig_temp_raw_c3_465,"eV",chord
mdsput, '\c3:offset_raw:error', build_string_1, $
        sig_offset_raw_465,"counts",chord
mdsput, '\c3:chisq_raw', build_string_1, $
        chi_raw,"counts",chord
mdsput, '\c3:comment','build_with_units($,"")', $
        'Plasma parameters found using 464.7418 nm,'+ $
        ' with C IV'
mdsput, '\c4','build_with_units(465.830,"nm")'
mdsput, '\c4:emiss_raw', build_string_1, $
        amp_raw_c4_465,"counts",chord
mdsput, '\c4:vel_raw', build_string_1, $
        vel_raw_c4_465,"m/s",chord
mdsput, '\c4:temp_raw', build_string_1, $
        temp_raw_c4_465,"eV",chord
mdsput, '\c4:emiss_raw:error', build_string_1, $
        sig_amp_raw_c4_465,"counts",chord
mdsput, '\c4:vel_raw:error', build_string_1, $
        sig_vel_raw_c4_465,"m/s",chord
mdsput, '\c4:temp_raw:error', build_string_1, $
        sig_temp_raw_c4_465,"eV",chord
mdsput, '\c4:comment','build_with_units($,"")', $
           'Plasma parameters found using 465.830 nm,'+ $
            ' with C III triplet'
for i=1,9 do $
   mdsput, '\c3:fit_raw_0'+stremo(i), $
          build_string_1,yfit_raw(*,i-1),"counts",wave_fit
```

```
      for i=10,20 do $
         mdsput, '\c3:fit_raw_'+stremo(i), $
                 build_string_1,yfit_raw(*,i-1),"counts",wave_fit
      for i=1,9 do $
         mdsput, '\c4:fit_raw_0'+stremo(i), $
                 build_string_1,yfit_raw(*,i-1),"counts",wave_fit
      for i=10,20 do $
         mdsput, '\c4:fit_raw_'+stremo(i), $
                 build_string_1,yfit_raw(*,i-1),"counts",wave_fit


      mdsclose
    endif
    nowrite:
endif else begin
  print,'Calculating c3_465 parameters'
  iccd_raw_multi_emis,intense,d_raw,wave,inst_line, $
      fwhm_inst,func_name=f_name(regions(i)), $
      lamb_ideal=all_ideal_wav(regions(i)), $
      mass=all_mass(regions(i)), $
      par_raw=par_raw,sig_raw=sig_raw,chi_raw=chi_raw, $
      amp_raw=amp_raw,sig_amp=sig_amp_raw, $
      vel_raw=vel_raw,sig_vel=sig_vel_raw, $
      temp_raw=temp_raw,sig_temp=sig_temp_raw, $
      offset_raw=offset_raw,sig_offset=sig_offset_raw, $
      wave_fit=wave_fit,yfit_raw=yfit_raw, $
      error_raw=error_raw
  write_raw_param,shot,all_ions(regions(i)), $
              all_ideal_wav(regions(i)), $
              amp_raw,vel_raw,temp_raw,offset_raw, $
              sig_amp_raw,sig_vel_raw, $
```

```
                    sig_temp_raw,sig_offset_raw, $
                    chi_raw,error_raw,wave_fit,yfit_raw
      endelse
    endif
    if n_elements(where(Number_lines(regions) eq -1)) lt 2 then begin
      print,'Calculating '+all_ions(regions(i))+' '+ $
            stremo(all_ideal_wav(regions(i)))+' parameters'
      iccd_raw_emis,intense,d_raw,wave,inst_line,fwhm_inst, $
                  lamb_ideal=all_ideal_wav(regions(i)), $
                  mass=all_mass(regions(i)), $
                  par_raw=par_raw,sig_raw=sig_raw,chi_raw=chi_raw, $
                  amp_raw=amp_raw,sig_amp=sig_amp_raw, $
                  vel_raw=vel_raw,sig_vel=sig_vel_raw, $
                  temp_raw=temp_raw,sig_temp=sig_temp_raw, $
                  offset_raw=offset_raw,sig_offset=sig_offset_raw, $
                  wave_fit=wave_fit,yfit_raw=yfit_raw, $
                  error_raw=error_raw
      write_raw_param,shot,all_ions(regions(i)), $
                    all_ideal_wav(regions(i)), $
                    amp_raw,vel_raw,temp_raw,offset_raw, $
                    sig_amp_raw,sig_vel_raw, $
                    sig_temp_raw,sig_offset_raw, $
                    chi_raw,error_raw,wave_fit,yfit_raw
    endif
  end
 endcase
endfor
blank:
!p.multi=0
end
```

iccd_raw_multi_emis.pro

```
pro iccd_raw_multi_emis,intense,d_raw,wave,inst_line,fwhm_inst, $
                func_name=func_name, $
                lamb_ideal=lamb_ideal,mass=mass,itmax=itmax,tol=tol, $
                par_raw=par_raw,sig_raw=sig_raw,yfit_raw=yfit_raw, $
                chi_raw=chi_raw, $
                amp_raw=amp_raw,sig_amp_raw=sig_amp_raw, $
                vel_raw=vel_raw,sig_vel=sig_vel_raw, $
                temp_raw=temp_raw,sig_temp=sig_temp_raw, $
                offset_raw=offset_raw,sig_offset=sig_offset_raw,ps=ps, $
                wave_fit=wave_fit,error_raw=error_raw

default,ps,0

n_el=n_elements(wave)
n_chord=n_elements(intense(0,*))
par_raw=fltarr(4,n_chord)
sig_raw=fltarr(4,n_chord)
yfit_raw=fltarr(n_el,n_chord)
chi_raw=fltarr(n_chord)
default,fwhm_inst,fltarr(n_chord)+(wave(1)-wave(0))
if n_elements(d_raw) eq 0 then begin
  d_raw=fltarr(n_chord)
  for i=0,n_chord-1 do begin
     err1=intense(0:19,i)-total(intense(0:19,i))/20
     err2=intense(n_el-20:n_el-1,i)-total(intense(n_el-20:n_el-1,i))/20
     d_raw(i)=sqrt(total(err1^2)/(19*sqrt(total(intense(0:19,i))/20))) > $
              sqrt(total(err2^2)/ $
              (19*sqrt(total(intense(n_el-20:n_el-1,i))/20)))>2
```

```
  endfor
endif
;
;Physical Constants
;
fwhm=fwhm_inst ;instrument FWHM wavelength
default,lamb_ideal,227.089 ;ideal waveleangth nm
default,mass,12.0107 ;mass C AMU
alpha_temp = mass * 936800890. ;gaussian constant = mc^2/e,
c = 299792458 ;speed of light in m/sec
fwhm=fwhm_inst*3/7          ;correcting fwhm to a(2) scale
tmin=(wave(1)-wave(0))*3/7 ;minimun FWHM 1 pixel
fwhm_temper=fwhm*tmin*7/3
lamb_ideal=lamb_ideal
x=wave

n_ideal=min(where(wave ge lamb_ideal))
x=wave(((n_ideal-40)>0):((n_ideal+120)<(n_el-1)))
n_x=n_elements(x)
yfit_raw=fltarr(n_x,n_chord)
wave_fit=x
error_raw=fltarr(n_chord)
;print,n_ideal,n_el,n_elements(intense(*,0)),n_elements(intense(0,*))

if ps ne 1 then window,17,XPOS=20 , YPOS=20 , XSIZE=1000 , YSIZE=800
!p.multi=[0,4,5]

a=fltarr(4)
for i=0,n_chord-1 do begin
   y=intense(((n_ideal-40)>0):((n_ideal+120)<(n_el-1)),i)
```

```
a1=inst_line(*,i)
w=fltarr(n_x)+1.;/(d_raw(i)*sqrt(y>1.))^2
yfitg=gaussfit(x(0:81<(n_x-1)),y(0:81<(n_x-1)),b,nterms=4)
a(0)=b(0)*b(2)*1.25
a(1)=b(1)
a(2)=sqrt((b(2)^2-fwhm_temper(i)^2)>tmin)
a(3)=b(3)
yfit=err_curvefit_wide(x,y,w,a,a1,sigma,chisq=chisq, $
                       function_name='f_emiss_wide_'+func_name, $
                       curve_err=curve_err,itmax=itmax, $
                       tol=tol,pder=pder)
par_raw(*,i)=a
yfit_raw(*,i)=yfit
error_raw(i)=curve_err
plot,x,y,title='chord '+stremo(i+1)
oplot,x,yfit,co=2
oplot,x(0:81<(n_x-1)),yfitg,co=3,linestyle=2
oplot,x,a1*max(y)/max(a1),co=4,linestyle=2
;
;correct error analysis
;
nterms=n_elements(a)
nfree = n_elements(y) - nterms ; Degrees of freedom
diag = lindgen(nterms)*(nterms+1) ; Subscripts of diagonal elements
Weights=fltarr(n_x)+1./(d_raw(i)*sqrt(y>1.))^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
chisq = total(Weights*(y-yfit)^2)/nfree     ; Current chi squared
```

```
   sig_raw(*,i)=sigma
   chi_raw(i)=chisq
endfor

amp_raw=transpose(par_raw(0,*))
sig_amp_raw=abs(transpose(sig_raw(0,*)))
offset_raw=transpose(par_raw(3,*))
sig_offset_raw=abs(transpose(sig_raw(3,*)))
vel_raw=(lamb_ideal-transpose(par_raw(1,*)))*c/ $
        (lamb_ideal*cos(35*!pi/180))
sig_vel_raw= transpose(sig_raw(1,*))*c/(lamb_ideal*cos(35*!pi/180))
del_lamb=abs(transpose(par_raw(2,*)))>tmin
temp_raw=alpha_temp*(del_lamb/lamb_ideal)^2
sig_temp_raw=2*alpha_temp*del_lamb*transpose(sig_raw(2,*))/lamb_ideal^2

if ps ne 1 then window,16,XPOS=20 , YPOS=20 , XSIZE=1000 , YSIZE=800
plot,findgen(n_chord)+1,amp_raw, $
     title='Amplitude Shot='+ $
     stremo( mdsvalue('$shotname', /quiet, status=status) ), $
     charsize=2,thick=2,background=1,color=0
plot,findgen(n_chord)+1,vel_raw,title='Velocity', $
     charsize=2,thick=2,background=1,color=0, $
     yrange=[(min(vel_raw)>(-2e5))<0,(max(vel_raw)<2e5)]
plot,findgen(n_chord)+1,temp_raw,title='Temperature', $
     charsize=2,thick=2,background=1,color=0, $
     yrange=[(min(temp_raw)>(-1))<0,(max(temp_raw)<1e3)]
plot,findgen(n_chord)+1,offset_raw,title='Offset', $
     charsize=2,thick=2,background=1,color=0
!p.multi=0
end
```

find_iccd_shell_param_SYIAH.pro

```
pro find_iccd_shell_param_SYIAH,shot
default,shot,30910022
ss,shot

;getting data
chord=findgen(20)+1
build_string_1 = 'build_signal(build_with_units($1,$2),' + $
'*,' + $
'$3,,)'

if not(data_there('\iccd_01')) then goto,blank
intense=data('\iccd_01',xaxis=wave)

; looking for lines
all_ions=['c3','c3','c4','c4','c5','c6','c6', $
          'n3','n3','n4','n4','n4','n5','n6','n6', $
          'o3','o3','o4','o5','o5',$
          'o6','o6','o7', $
          'He2', $
          'b3']
all_mass=[12.107  ,12.107  ,12.107  ,12.107  ,12.107  ,12.107  ,12.107  , $
          14.00674,14.00674,14.00674,14.00674,14.00674,14.00674,14.00674, $
          14.00674,15.9994 ,15.9994 ,15.9994 ,15.9994 ,15.9994 , $
          15.9994 ,15.9994 ,15.9994 , $
          4.002602, $
          10.811 ]
all_ideal_wav=[229.687 ,464.7418,465.83 ,580.131,227.089,343.369,529.1 , $
               268.218 ,409.736 ,638.075,264.562,347.872,298.078,205.61, $
```

```
                289.64  ,326.0857,375.4696,306.343,559.789,278.101, $
                381.135 ,383.424 ,255.9, $
                468.57  , $
                206.5776]
max_ideal_wav=[229.687 ,465.1473,465.83 ,581.197,227.792,343.369,529.1 , $
                268.218 ,410.339 ,638.075,264.698,348.300,299.843,205.61, $
                289.64  ,326.7204,375.9875,307.160,559.789,278.985, $
                381.135 ,383.424 ,255.9, $
                468.57  , $
                206.7233]
Number_lines=[1,-1,-1, 2, 3, 1, 1, $
              1, 2, 1, 3, 3, 3, 1, 1, $
              3, 3, 2, 1, 3, $
              1, 1, 1, $
              1, $
              2]  ; use -1 for the c III c IV region
f_name=['','three_cIII_465','','two_cIV_580' ,'three_cV_227' ,'','', $
        '','two_nIII_409'  ,'','three_nIV_264','three_nIV_347', $
        'three_nV_298','','','three_oIII_326','three_oIII_375', $
        'two_oIV_306','','three_oV_278' ,'','','','','two_bIII_206']

n_ions=n_elements(all_ions)
n_start=intarr(n_ions)

for i=0,n_ions-1 do n_start(i)=min(where(wave ge all_ideal_wav(i)))

regions=where(n_start ge 5)
if regions(0) lt 0 then goto,blank
n_regions=n_elements(regions)
use_region=intarr(n_regions)
```

```
for i=0,n_regions-1 do $
   use_region(i)=(data_there('\'+all_ions(regions(i))+':emiss_raw'))
n_usefull=where(use_region eq 1)
if n_usefull(0) lt 0 then goto,blank

regions=regions(n_usefull)
n_regions=n_elements(regions)

shellfit=intarr(n_regions)
a_save=fltarr(4,n_regions)
window,10
!p.multi=[0,1,n_regions]
for i=0,n_regions-1 do begin
   emiss=data('\'+all_ions(regions(i))+':emiss_raw',xaxis=chord)
   chord=chord(where(finite(emiss) eq 1))
   emiss=(emiss(where(finite(emiss) eq 1))<1e10)>(-1e10)
   yfit=gaussfit(chord,emiss,a,nterms=4)
   a_save(*,i)=a
   plot,chord,emiss
   oplot,chord,yfit,co=2
   if (((a(1) + abs(a(2))*7./6.) lt 20) and $
      ((a(1) - abs(a(2))*7./6.) gt 1)) and  $
      (a(0) gt 100>2*abs(a(3))) and $
      (Number_lines(regions(i)) ne -1) $
      then shellfit(i)=1
   ;stop
endfor
print,shellfit
n_shellfit=where(shellfit eq 1)
if n_shellfit(0) lt 0 then goto,blank
```

```
regions=regions(n_shellfit)
a_save=a_save(*,n_shellfit)
n_regions=n_elements(regions)
print,regions,all_ions(regions)

view=mdsvalue('.zapanalysis.iccd.spectra:view')
intense=data('\iccd_01',xaxis=wave)
intense=data('\iccd_01:inst_func',xaxis=pix)
n_el=n_elements(wave)
n_inst=n_elements(pix)
intense=fltarr(n_el,20)
inst_line=fltarr(n_inst,20)
inst_fwhm=fltarr(20)
for i=0,8 do begin
   intense(*,i)=data('\iccd_0'+stremo(i+1))>(-1.e3)
   inst_line(*,i)=data('\iccd_0'+stremo(i+1)+':inst_func')
   inst_fwhm(i)=data('\iccd_0'+stremo(i+1)+':inst_fwhm')
endfor
for i=9,19 do begin
   intense(*,i)=data('\iccd_'+stremo(i+1))>(-1.e3)
   inst_line(*,i)=data('\iccd_'+stremo(i+1)+':inst_func')
   inst_fwhm(i)=data('\iccd_'+stremo(i+1)+':inst_fwhm')
endfor
fwhm_pix = inst_fwhm
wave_s=wave
intense_s=intense
inst_line_s=inst_line
inst_fwhm_s=inst_fwhm

l_edge_ave=fltarr(20)
```

```
r_edge_ave=fltarr(20)
l_edge_sigma=fltarr(20)
r_edge_sigma=fltarr(20)
for i=0,19 do begin
   l_edge_ave(i)=total(intense(1:10,i))/10
   r_edge_ave(i)=total(intense(500:509,i))/10
   l_edge_sigma(i)=sqrt(total((intense(1:10,i)-l_edge_ave(i))^2)/9)
   r_edge_sigma(i)=sqrt(total((intense(500:509,i)-r_edge_ave(i))^2)/9)
endfor

d_raw=((l_edge_sigma/(sqrt(l_edge_ave>1.)))<( $
       r_edge_sigma/(sqrt(r_edge_ave>1.))))>1.
ave_edge=(l_edge_ave+r_edge_ave)/2

window,0
!p.multi=0
plot,wave,intense(*,0),yra=[-.01,1]*max(intense)
for i=0,19 do oplot,wave,intense(*,i),co=2+i
hline,5*max(d_raw)

for i=0,n_regions-1 do begin
   ss,shot
   inst_fwm=fwhm_pix
   wave=wave_s
   intense=intense_s
   inst_line=inst_line_s
   inst_fwhm=inst_fwhm_s
   if Number_lines(regions(i)) eq 1 then $
     funct_name='f_emiss_wide' $
   else $
```

```
  funct_name='f_emiss_wide_'+stremo(f_name(regions(i)))
;stop
find_center_SYIAH,intense,wave,d_raw,inst_line,pix,inst_fwhm, $
                  a_save(1,i),a_save(2,i)/4., $
                  all_ideal_wav(regions(i)), $
                  max_ideal_wav(regions(i)), $
                  all_mass(regions(i)),funct_name, $
                  all_ions(regions(i)), $
                  r_chord=r_chord, $
                  amp_shell_l=amp_shell_l, $
                  sig_amp_shell_l=sig_amp_shell_l, $
                  vel_shell_l=vel_shell_l, $
                  sig_vel_shell_l=sig_vel_shell_l, $
                  temp_shell_l=temp_shell_l, $
                  sig_temp_shell_l=sig_temp_shell_l, $
                  offset_shell_l=offset_shell_l, $
                  sig_offset_shell_l=sig_offset_shell_l, $
                  amp_shell_r=amp_shell_r, $
                  sig_amp_shell_r=sig_amp_shell_r, $
                  vel_shell_r=vel_shell_r, $
                  sig_vel_shell_r=sig_vel_shell_r, $
                  temp_shell_r=temp_shell_r, $
                  sig_temp_shell_r=sig_temp_shell_r, $
                  offset_shell_r=offset_shell_r,
                  sig_offset_shell_r=sig_offset_shell_r, $
                  chi_1=chi_1, $
                  center=center,x_error=x_error,error=error, $
                  yfit_l=yfit_l,yfit_r=yfit_r,fit_wave=fit_wave

write_shell_param,shot,all_ions(regions(i)),all_ideal_wav(regions(i)), $
```

```
                    center,r_chord,x_error,error, $
                    amp_shell_l,vel_shell_l, $
                    temp_shell_l,offset_shell_l, $
                    sig_amp_shell_l,sig_vel_shell_l, $
                    sig_temp_shell_l,sig_offset_shell_l, $
                    amp_shell_r,vel_shell_r, $
                    temp_shell_r,offset_shell_r, $
                    sig_amp_shell_r,sig_vel_shell_r,sig_temp_shell_r, $
                    sig_offset_shell_r, $
                    fit_wave,yfit_l,yfit_r

endfor
blank:
end
```

find_center_SYIAH.pro

```
pro find_center_SYIAH,intense,wave,d_raw,inst_line,pix,inst_fwhm, $
                      start_cen,d_center,lamb_ideal,lamb_max,mass, $
                      function_name,ion, $
                      r_chord=r_chord, $
                      amp_shell_l=amp_shell_l, $
                      sig_amp_shell_l=sig_amp_shell_l, $
                      vel_shell_l=vel_shell_l, $
                      sig_vel_shell_l=sig_vel_shell_l, $
                      temp_shell_l=temp_shell_l, $
                      sig_temp_shell_l=sig_temp_shell_l, $
                      offset_shell_l=offset_shell_l, $
                      sig_offset_shell_l=sig_offset_shell_l, $
                      amp_shell_r=amp_shell_r, $
                      sig_amp_shell_r=sig_amp_shell_r, $
                      vel_shell_r=vel_shell_r, $
                      sig_vel_shell_r=sig_vel_shell_r, $
                      temp_shell_r=temp_shell_r, $
                      sig_temp_shell_r=sig_temp_shell_r, $
                      offset_shell_r=offset_shell_r, $
                      sig_offset_shell_r=sig_offset_shell_r, $
                      chi_1=chi_1, $
                      center=center,x_error=x_error,error=error, $
                      yfit_l=yfit_l,yfit_r=yfit_r,fit_wave=fit_wave

n_el=n_elements(wave)
n_inst=n_elements(pix)

shot = mdsvalue('$shot', /quiet, status=status)
```

```
if ((shot le 21007000) or (shot ge 500000000)) then $
  d_r=1.24 $
else $
  d_r=1.78

x=start_cen+d_center*(findgen(11)-5)
i_use=where((x ge 2) and (x le 18))
if i_use(0) ne -1 then $
  x=x(i_use) $
else $
  x=[2.,6.,10.,14.,18.]
n_x=n_elements(x)
if n_x lt 5 then begin
  x=[2.,6.,10.,14.,18.]
  n_x=5
endif

amp_shell_l_s=fltarr(20,n_x)
amp_shell_r_s=fltarr(20,n_x)
vel_shell_l_s=fltarr(20,n_x)
vel_shell_r_s=fltarr(20,n_x)
temp_shell_l_s=fltarr(20,n_x)
temp_shell_r_s=fltarr(20,n_x)
offset_shell_l_s=fltarr(20,n_x)
offset_shell_r_s=fltarr(20,n_x)
r_chord_20_s=fltarr(20,n_x)

for i=0,n_x-1 do begin
   center=x(i)
```

```
shell_par_SYIAH,wave,intense,d_raw,inst_line,inst_fwhm, $;data
                center,d_r, $;geometry
                lamb_ideal,lamb_max,mass,function_name, $;ion
                itmax=itmax,tol=tol, $;fit parameters
                r_chord=r_chord, $
                amp_shell_l=amp_shell_l, $
                sig_amp_shell_l=sig_amp_shell_l, $
                vel_shell_l=vel_shell_l, $
                sig_vel_shell_l=sig_vel_shell_l, $
                temp_shell_l=temp_shell_l, $
                sig_temp_shell_l=sig_temp_shell_l, $
                offset_shell_l=offset_shell_l, $
                sig_offset_shell_l=sig_offset_shell_l, $
                amp_shell_r=amp_shell_r, $
                sig_amp_shell_r=sig_amp_shell_r, $
                vel_shell_r=vel_shell_r, $
                sig_vel_shell_r=sig_vel_shell_r, $
                temp_shell_r=temp_shell_r, $
                sig_temp_shell_r=sig_temp_shell_r, $
                offset_shell_r=offset_shell_r, $
                sig_offset_shell_r=sig_offset_shell_r, $
                chi_1=chi_1, $
                par_shell_l=par_shell_l,par_shell_r=par_shell_r, $
                l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
                l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r
amp_shell_l_s(*,i)=amp_shell_l
amp_shell_r_s(*,i)=amp_shell_r
vel_shell_l_s(*,i)=vel_shell_l
vel_shell_r_s(*,i)=vel_shell_r
temp_shell_l_s(*,i)=temp_shell_l
```

```
   temp_shell_r_s(*,i)=temp_shell_r
   offset_shell_l_s(*,i)=offset_shell_l
   offset_shell_r_s(*,i)=offset_shell_r
   r_chord_20_s(*,i)=r_chord
endfor

chi2_amp=fltarr(n_x)
chi2_vel=fltarr(n_x)
for i=0,n_x-1 do begin
   n=where((abs(r_chord_20_s(*,i)) le 2*d_r) and $
           (amp_shell_r_s(*,i) gt 0) and $
           (amp_shell_l_s(*,i) gt 0))
   if n_elements(n) le 1 then begin
     chi2_amp(i)=100
     chi2_vel(i)=100
   endif else begin
     chi2_amp(i)=total(((amp_shell_r_s(n,i)-amp_shell_l_s(n,i))/ $
                        ((amp_shell_r_s(n,i)+amp_shell_l_s(n,i))>1))^2)
     chi2_vel(i)=total(((vel_shell_r_s(n,i)-vel_shell_l_s(n,i))/ $
                        (abs(vel_shell_r_s(n,i)+vel_shell_l_s(n,i))>1e4))^2)
     chi2_amp(i)=sqrt(chi2_amp(i)/n_elements(n))
     chi2_vel(i)=sqrt(chi2_vel(i)/n_elements(n))
   endelse
   plot,r_chord_20_s(*,i),amp_shell_r_s(*,i)
   oplot,r_chord_20_s(*,i),amp_shell_l_s(*,i),co=2
   plot,r_chord_20_s(*,i),vel_shell_r_s(*,i)
   oplot,r_chord_20_s(*,i),vel_shell_l_s(*,i),co=2
endfor
plot,x,chi2_amp,yra=[0,1]
plot,x,chi2_vel,yra=[0,1]
```

```
error=sqrt(chi2_amp^2+chi2_vel^2)

print,'center location=',x(min(where(error eq min(error))))

min_dx=d_center
count=0
while (min_dx ge .01) and count le 20 do begin
   n_x=n_elements(x)
   n_min=min(where(error(2:n_x-3) eq min(error(2:n_x-3))))+2
   i_1=indgen(4)-2+n_min
   x=[x,(x(i_1)+x(i_1+1))*0.5]

   for i=n_x,n_x+3 do begin
      center=x(i)
      shell_par_SYIAH,wave,intense,d_raw,inst_line,inst_fwhm, $;data
                    center,d_r, $;geometry
                    lamb_ideal,lamb_max,mass,function_name, $;ion
                    itmax=itmax,tol=tol, $;fit parameters
                    r_chord=r_chord, $
                    amp_shell_l=amp_shell_l, $
                    sig_amp_shell_l=sig_amp_shell_l, $
                    vel_shell_l=vel_shell_l, $
                    sig_vel_shell_l=sig_vel_shell_l, $
                    temp_shell_l=temp_shell_l, $
                    sig_temp_shell_l=sig_temp_shell_l, $
                    offset_shell_l=offset_shell_l, $
                    sig_offset_shell_l=sig_offset_shell_l, $
                    amp_shell_r=amp_shell_r, $
                    sig_amp_shell_r=sig_amp_shell_r, $
```

```
                vel_shell_r=vel_shell_r, $
                sig_vel_shell_r=sig_vel_shell_r, $
                temp_shell_r=temp_shell_r, $
                sig_temp_shell_r=sig_temp_shell_r, $
                offset_shell_r=offset_shell_r, $
                sig_offset_shell_r=sig_offset_shell_r, $
                chi_1=chi_1, $
                par_shell_l=par_shell_l,par_shell_r=par_shell_r, $
                l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
                l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r
amp_shell_l_s=[[amp_shell_l_s],[amp_shell_l]]
amp_shell_r_s=[[amp_shell_r_s],[amp_shell_r]]
vel_shell_l_s=[[vel_shell_l_s],[vel_shell_l]]
vel_shell_r_s=[[vel_shell_r_s],[vel_shell_r]]
temp_shell_l_s=[[temp_shell_l_s],[temp_shell_l]]
temp_shell_r_s=[[temp_shell_r_s],[temp_shell_r]]
offset_shell_l_s=[[offset_shell_l_s],[offset_shell_l]]
offset_shell_r_s=[[offset_shell_r_s],[offset_shell_r]]
r_chord_20_s=[[r_chord_20_s],[r_chord]]

n=where((abs(r_chord_20_s(*,i)) le 2*d_r) and $
        (amp_shell_r_s(*,i) gt 0) and $
        (amp_shell_l_s(*,i) gt 0))
if n(0) eq -1 then begin
  chi2_amp=[chi2_amp,100]
  chi2_vel=[chi2_vel,100]
endif else begin
  chi2_amp=[chi2_amp,total(((amp_shell_r_s(n,i)- $
                             amp_shell_l_s(n,i))/ $
                          ((amp_shell_r_s(n,i)+ $
```

```
                                amp_shell_l_s(n,i))>1))^2)]
      chi2_vel=[chi2_vel,total(((vel_shell_r_s(n,i)- $
                                 vel_shell_l_s(n,i))/ $
                     (abs(vel_shell_r_s(n,i)+ $
                          vel_shell_l_s(n,i))>1e4))^2)]
      chi2_amp(i)=sqrt(chi2_amp(i)/n_elements(n))
      chi2_vel(i)=sqrt(chi2_vel(i)/n_elements(n))
   endelse
   plot,r_chord_20_s(*,i),amp_shell_r_s(*,i)
   oplot,r_chord_20_s(*,i),amp_shell_l_s(*,i),co=2
   plot,r_chord_20_s(*,i),vel_shell_r_s(*,i)
   oplot,r_chord_20_s(*,i),vel_shell_l_s(*,i),co=2
endfor
n_x=n_x+4
error=sqrt(chi2_amp^2+chi2_vel^2)
i_sort=sort(x)
x=x(i_sort)
error=error(i_sort)
chi2_amp=chi2_amp(i_sort)
chi2_vel=chi2_vel(i_sort)
amp_shell_l_s=amp_shell_l_s(*,i_sort)
amp_shell_r_s=amp_shell_r_s(*,i_sort)
vel_shell_l_s=vel_shell_l_s(*,i_sort)
vel_shell_r_s=vel_shell_r_s(*,i_sort)
temp_shell_l_s=temp_shell_l_s(*,i_sort)
temp_shell_r_s=temp_shell_r_s(*,i_sort)
offset_shell_l_s=offset_shell_l_s(*,i_sort)
offset_shell_r_s=offset_shell_r_s(*,i_sort)
r_chord_20_s=r_chord_20_s(*,i_sort)
```

```
   min_dx=min(x(1:n_x-1)-x(0:n_x-2))
   count=count+1
   ;stop
endwhile

window,30,XPOS=40 , YPOS=40 , XSIZE=1000 , YSIZE=800
!p.multi=[0,1,3]
plot,x,chi2_amp,yra=[0,1]
plot,x,chi2_vel,yra=[0,1]

error=sqrt(chi2_amp^2+chi2_vel^2)
n_x=n_elements(x)
plot,x,error,yra=[0,1]
i=min(where(error(2:n_x-3) eq min(error(2:n_x-3))))+2
print,'center location=',x(i)

!p.multi=[0,2,2]
window,31,XPOS=20 , YPOS=20 , XSIZE=1000 , YSIZE=800

vel_raw=data('\'+ion+':vel_raw')
temp_raw=data('\'+ion+':temp_raw')
amp_raw=data('\'+ion+':emiss_raw')
offset_raw=data('\'+ion+':offset_raw')

plot,r_chord_20_s(*,i),vel_raw,title='vel  shot='+stremo(shot)
oplot,r_chord_20_s(*,i),vel_shell_l_s(*,i),color=4
oplot,r_chord_20_s(*,i),vel_shell_r_s(*,i),color=2
plot,r_chord_20_s(*,i),temp_raw,title='temp'
oplot,r_chord_20_s(*,i),temp_shell_l_s(*,i),color=4
oplot,r_chord_20_s(*,i),temp_shell_r_s(*,i),color=2
```

```
plot,r_chord_20_s(*,i),10*amp_raw/max(r_chord_20_s(*,i)),title='amp'
oplot,r_chord_20_s(*,i),amp_shell_l_s(*,i),color=4
oplot,r_chord_20_s(*,i),amp_shell_r_s(*,i),color=2
plot,r_chord_20_s(*,i),10*offset_raw/max(r_chord_20_s(*,i)),title='offset'
oplot,r_chord_20_s(*,i),offset_shell_l_s(*,i),color=4
oplot,r_chord_20_s(*,i),offset_shell_r_s(*,i),color=2

!p.multi=0
center=x(i)
shell_par_SYIAH,wave,intense,d_raw,inst_line,inst_fwhm, $;data
                center,d_r, $;geometry
                lamb_ideal,lamb_max,mass,function_name, $;ion
                itmax=itmax,tol=tol, $;fit parameters
                r_chord=r_chord, $
                amp_shell_l=amp_shell_l, $
                sig_amp_shell_l=sig_amp_shell_l, $
                vel_shell_l=vel_shell_l, $
                sig_vel_shell_l=sig_vel_shell_l, $
                temp_shell_l=temp_shell_l, $
                sig_temp_shell_l=sig_temp_shell_l, $
                offset_shell_l=offset_shell_l, $
                sig_offset_shell_l=sig_offset_shell_l, $
                amp_shell_r=amp_shell_r, $
                sig_amp_shell_r=sig_amp_shell_r, $
                vel_shell_r=vel_shell_r, $
                sig_vel_shell_r=sig_vel_shell_r, $
                temp_shell_r=temp_shell_r, $
                sig_temp_shell_r=sig_temp_shell_r, $
                offset_shell_r=offset_shell_r, $
                sig_offset_shell_r=sig_offset_shell_r, $
```

```
                chi_1=chi_1, $
                par_shell_l=par_shell_l,par_shell_r=par_shell_r, $
                l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
                l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r, $
                yfit_l=yfit_l,yfit_r=yfit_r,fit_wave=fit_wave
x_error=x
end
```

shell_par_SYIAH.pro

```
pro shell_par_SYIAH,wave,intense,d_raw,inst_line,fwhm_pix, $;data
                    center,d_r, $;geometry
                    lamb_ideal,lamb_max,mass,function_name, $;ion
                    itmax=itmax,tol=tol, $;fit parameters
                    r_chord=r_chord,shell_max=shell_max, $
                    cen_per=cen_per, $;returned geomtery
                    err_center=err_center,err_rmax=err_rmax, $
                    err_cen_per=err_cen_per, $;returned errors
                    amp_shell_l=amp_shell_l, $
                    sig_amp_shell_l=sig_amp_shell_l, $
                    vel_shell_l=vel_shell_l, $
                    sig_vel_shell_l=sig_vel_shell_l, $
                    temp_shell_l=temp_shell_l, $
                    sig_temp_shell_l=sig_temp_shell_l, $
                    offset_shell_l=offset_shell_l, $
                    sig_offset_shell_l=sig_offset_shell_l, $
                    amp_shell_r=amp_shell_r, $
                    sig_amp_shell_r=sig_amp_shell_r, $
                    vel_shell_r=vel_shell_r, $
                    sig_vel_shell_r=sig_vel_shell_r, $
                    temp_shell_r=temp_shell_r, $
                    sig_temp_shell_r=sig_temp_shell_r, $
                    offset_shell_r=offset_shell_r, $
                    sig_offset_shell_r=sig_offset_shell_r, $
                    chi_1=chi_1, $
                    par_shell_l=par_shell_l,par_shell_r=par_shell_r, $
                    l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
                    l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r, $
```

```
                    yfit_l=yfit_l,yfit_r=yfit_r,fit_wave=fit_wave

d_raw=d_raw>1.
default,tol,1e-5
default,chi_1,0
;background level for 726025
default,background,[3481.25426 ,3581.75088,3937.5536 , $
                    3335.32269 ,3170.51358, $
                    3046.82294 ,3248.75191,2924.01434, $
                    2709.362648,3002.805,   $
                    2884.00000 ,2829.1095 ,2957.00886, $
                    2707.26522 ,2786.80752, $
                    2949.16708 ,3117.4825 ,2963.0286 , $
                    3219.05781 ,3556.84632]

;
;Physical Constants
;
fwhm=fwhm_pix                ;instrument FWHM wavelength
alpha_temp = mass * 936800890. ;gaussian constant = mc^2/e,
c = 299792458. ;speed of light in m/sec
fwhm_temper=fwhm*3/7 ;correcting fwhm to a(2) scale
fwhm_temper=fwhm_temper*(wave(1)-wave(0))
tmin=(wave(1)-wave(0))*3/7 ;minimun FWHM 1 pixel

;finding region of interest
n_el=n_elements(wave)
nid_min=min(where(wave ge lamb_ideal))
nid_max=min(where(wave ge lamb_max))
wave=wave(((nid_min-60)>0):((nid_max+40)<(n_el-1)))
```

```
intense=intense(((nid_min-60)>0):((nid_max+40)<(n_el-1)),*)

n_el=n_elements(wave)
x=wave
fit_wave=x

;Initializing arrays
par_shell_l=dblarr(4,20) ;parameters from the left fit
sig_shell_l_all=dblarr(4,20) ;errors from the left fit
sig_shell_l_data=dblarr(4,20) ;errors from the left fit due to iccd
sig_shell_l_geom=dblarr(4,20) ;errors from the left fit due to center
                                    ; location
par_shell_r=dblarr(4,20) ;parameters from the right fit
sig_shell_r_all=dblarr(4,20)     ;errors from the right fit
sig_shell_r_data=dblarr(4,20) ;errors from the right fit due to iccd
sig_shell_r_geom=dblarr(4,20) ;errors from the right fit due to center
                                    ; location
yfit_l=fltarr(n_el,20)              ;fits from the left
yfit_r=fltarr(n_el,20)              ;fits from the right

;determining the L matrix
default,d_r,1.24 ;distance between the chords
default,center,10.5 ;center location of the plasma in chord units
default,err_center,.25 ;error in the center location (chord units)
;shell_max ;maximun shell radius
default,err_rmax,1. ;error in the maximun shell radius (chord units)
default,cen_per,.75 ;location of chord i in shell i (from 0 to 1)
;1.0 puts chord r of shell i-1
default,err_cen_per,.25 ;error in the location of chord i in shell i
```

```
shell_length_mid,center=center,d_r=d_r,shell_max=shell_max, $
              r_chord=r_chord,err_center=err_center,err_rmax=err_rmax, $
              cen_per=cen_per,err_cen_per=err_cen_per, $
              l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
              l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r, $
              err_cen_l_shell_l=err_cen_l_shell_l, $
              err_cen_l_shell_r=err_cen_l_shell_r, $
              err_max_l_shell_l=err_max_l_shell_l, $
              err_max_l_shell_r=err_max_l_shell_r, $
              err_per_l_shell_l=err_per_l_shell_l, $
              err_per_l_shell_r=err_per_l_shell_r, $
              n_shell_r=n_shell_r
;converting shells to ovals
l_shell_l=l_shell_l*2/sin(35*!pi/180)
l_shell_r=l_shell_r*2/sin(35*!pi/180)
err_l_shell_l=err_l_shell_l*2/sin(35*!pi/180)
err_l_shell_r=err_l_shell_r*2/sin(35*!pi/180)
err_cen_l_shell_l=err_cen_l_shell_l*2/sin(35*!pi/180)
err_cen_l_shell_r=err_cen_l_shell_r*2/sin(35*!pi/180)
err_max_l_shell_l=err_max_l_shell_l*2/sin(35*!pi/180)
err_max_l_shell_r=err_max_l_shell_r*2/sin(35*!pi/180)
err_per_l_shell_l=err_per_l_shell_l*2/sin(35*!pi/180)
err_per_l_shell_r=err_per_l_shell_r*2/sin(35*!pi/180)
;stop
;finding the inner most shell for each chord
first_shell_l=intarr(20)
first_shell_r=intarr(20)
for i=0,19 do $
   first_shell_l(i)=max(where(l_shell_l(i,*) gt 0.0001))
for i=0,19 do $
```

```
   first_shell_r(i)=min(where(l_shell_r(i,*) gt 0.0001))

;opening window for check of the fits
  window,20,XPOS=20 , YPOS=20 , XSIZE=1000 , YSIZE=800
  !p.multi=[0,4,5]

;fitting data from the left
for i=0,19 do begin
   ;initalizing arrays for cord i
   y_sub=dblarr(n_el)
   err_y_sub=dblarr(n_el)
   err_y_sub_data=dblarr(n_el)
   err_y_sub_geom=dblarr(n_el)
   a1=inst_line(*,i)
   ;calculating subtraction array
   for j=0,first_shell_l(i)-1 do begin
      a=par_shell_l(*,j)
      CALL_PROCEDURE,function_name, x, a, a1, f, pder
      y_sub=y_sub+f*l_shell_l(i,j)
      ;all sigmas
      sigma=sig_shell_l_all(*,j)
      err_f=sqrt(pder^2#sigma^2)
      err_y_sub=err_y_sub+(err_f^2*l_shell_l(i,j)^2+ $
                           f^2*err_l_shell_l(i,j)^2)
      ;error from data
      sigma=sig_shell_l_data(*,j)
      err_f=sqrt(pder^2#sigma^2)
      err_y_sub_data=err_y_sub_data+err_f^2*l_shell_l(i,j)^2
      ;error from geometry
      sigma=sig_shell_l_geom(*,j)
```

```
   err_f=sqrt(pder^2#sigma^2)
   err_y_sub_geom=err_y_sub_geom+(err_f^2*l_shell_l(i,j)^2+ $
                                 f^2*err_l_shell_l(i,j)^2)
endfor
err_y_sub=sqrt(err_y_sub)<y_sub
err_y_sub_data=sqrt(err_y_sub_data)<y_sub
err_y_sub_geom=sqrt(err_y_sub_geom)<y_sub

;calculating the waveform to be fitted
y=((intense(*,i)-y_sub)/l_shell_l(i,first_shell_l(i)))
y=y>0.0
err_intense=sqrt((sqrt(intense(*,i)>1)*d_raw(i))^2+background(i))
err_y=sqrt((err_intense^2+err_y_sub^2)/ $
            l_shell_l(i,first_shell_l(i))^2+ $
            (y*err_l_shell_l(i,first_shell_l(i))/ $
             l_shell_l(i,first_shell_l(i)))^2)> $
              d_raw(i)/l_shell_l(i,first_shell_l(i))

err_y_data=sqrt((err_intense^2+err_y_sub_data^2)/ $
                l_shell_l(i,first_shell_l(i))^2)
err_y_geom=sqrt((err_y_sub_geom^2/l_shell_l(i,first_shell_l(i))^2)+ $
                (y*err_l_shell_l(i,first_shell_l(i))/ $
                l_shell_l(i,first_shell_l(i)))^2)
if max(y) gt 1.*max(err_y) then begin
a1=inst_line(*,i)
w=fltarr(n_el)+1.
;fitting the data
yfit=gaussfit( x, y, a, NTERMS=4)
a(0)=a(0)*a(2)
a(2)=sqrt((a(2)^2-fwhm_temper(i)^2)>fwhm_temper(i)^2)
```

```
yfit=err_CURVEFIT_wide(x,y,w,a,a1,sigma,chisq=chisq, $
                       function_name=function_name, $
                       curve_err=curve_err,itmax=itmax, $
                       tol=tol,pder=pder)
;finding the errors
nterms=n_elements(a)
diag = lindgen(nterms)*(nterms+1) ; Subscripts of diagonal elements
Weights=1/err_y^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
par_shell_l(*,i)=a
sig_shell_l_all(*,i)=sigma
Weights=1/err_y_data^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] )
sig_shell_l_data(*,i)=sigma
Weights=1/err_y_geom^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
sig_shell_l_geom(*,i)=sigma

endif else begin
  yfit=dblarr(n_el)
  par_shell_l(*,i)=[0.0, lamb_ideal, tmin, 0.0]
```

```
   endelse
   ;plotting data
   plot,x,intense(*,i)/l_shell_l(i,first_shell_l(i)), $
          title='Chord '+stremo(i+1)+' '+stremo(r_chord(i)), $
          xra=[x(n_el/4),x(n_el*3/4)]
   oplot,x,(intense(*,i)+err_intense)/l_shell_l(i,first_shell_l(i)), $
         linestyle=1
   oplot,x,(intense(*,i)-err_intense)/l_shell_l(i,first_shell_l(i)), $
         linestyle=1
   oplot,x,yfit+y_sub/l_shell_l(i,first_shell_l(i)),color=4,linestyle=2
   oplot,x,y_sub/l_shell_l(i,first_shell_l(i)),color=3
   oplot,x,(y_sub+err_y_sub)/l_shell_l(i,first_shell_l(i)), $
         color=3,linestyle=2
   oplot,x,(y_sub-err_y_sub)/l_shell_l(i,first_shell_l(i)), $
         color=3,linestyle=2
   oplot,x,y,co=5,linestyle=2
   oplot,x,yfit,co=6,linestyle=1
   vline,lamb_ideal,color=2
   if (i eq 9) or (i eq 19) then begin
     xyouts,.85,.94,'Experimental spectra',/normal,charsize=.5
     xyouts,.85,.92,'Fitted spectra',/normal,color=4,charsize=.5
     xyouts,.85,.9,'Ideal wavelength',/normal,color=2,charsize=.5
   endif
   yfit_l(*,i)=y_sub+yfit*l_shell_l(i,first_shell_l(i))
endfor

;opening window for check of the fits
  window,21,XPOS=20 , YPOS=20 , XSIZE=1000 , YSIZE=800
  !p.multi=[0,4,5]
```

```
;fitting data from the right
for i=19,0,-1 do begin
   ;initalizing arrays for cord i
   y_sub=dblarr(n_el)
   err_y_sub=dblarr(n_el)
   err_y_sub_data=dblarr(n_el)
   err_y_sub_geom=dblarr(n_el)
   a1=inst_line(*,i)
   ;calculating subtraction array
   for j=n_shell_r-1,first_shell_r(i)+1,-1 do begin
      a=par_shell_r(*,j+(20-n_shell_r))
      CALL_PROCEDURE,function_name, x, a, a1, f, pder
      y_sub=y_sub+f*l_shell_r(i,j)
      sigma=sig_shell_r_all(*,j+(20-n_shell_r))
      err_f=sqrt(pder^2#sigma^2)
      err_y_sub=err_y_sub+(err_f^2*l_shell_r(i,j)^2+ $
                           f^2*err_l_shell_r(i,j)^2)
      sigma=sig_shell_r_data(*,j+(20-n_shell_r))
      err_f=sqrt(pder^2#sigma^2)
      err_y_sub_data=err_y_sub_data+err_f^2*l_shell_r(i,j)^2
      sigma=sig_shell_r_geom(*,j+(20-n_shell_r))
      err_f=sqrt(pder^2#sigma^2)
      err_y_sub_geom=err_y_sub_geom+(err_f^2*l_shell_r(i,j)^2+ $
                                     f^2*err_l_shell_r(i,j)^2)
   endfor
   err_y_sub=sqrt(err_y_sub)<y_sub
   err_y_sub_data=sqrt(err_y_sub_data)<y_sub
   err_y_sub_geom=sqrt(err_y_sub_geom)<y_sub
   ;calculating the waveform to be fitted
   y=((intense(*,i)-y_sub)/l_shell_r(i,first_shell_r(i)))
```

```
y=y>0.0
err_intense=sqrt((sqrt(intense(*,i)>1)*d_raw(i))^2+background(i))
err_y=sqrt((err_intense^2+err_y_sub^2)/ $
            l_shell_r(i,first_shell_r(i))^2+ $
            (y*err_l_shell_r(i,first_shell_r(i))/ $
             l_shell_r(i,first_shell_r(i)))^2)> $
              d_raw(i)/l_shell_l(i,first_shell_l(i))
err_y_data=sqrt((err_intense^2+err_y_sub_data^2)/ $
                 l_shell_r(i,first_shell_r(i))^2)> $
                  d_raw(i)/l_shell_l(i,first_shell_l(i))
err_y_geom=sqrt((err_y_sub_geom^2)/l_shell_r(i,first_shell_r(i))^2+ $
               (y*err_l_shell_r(i,first_shell_r(i))/ $
                l_shell_r(i,first_shell_r(i)))^2)
if max(y) gt 1.*max(err_y) then begin
yfit=gaussfit( x, y, a, NTERMS=4)
a(0)=a(0)*a(2)
a(2)=sqrt((a(2)^2-fwhm_temper(i)^2)>fwhm_temper(i)^2)
yfit=err_CURVEFIT_wide(x,y,w,a,a1,sigma,chisq=chisq, $
                       function_name=function_name, $
                       curve_err=curve_err,itmax=itmax, $
                       tol=tol,pder=pder)
;finding the errors
nterms=n_elements(a)
diag = lindgen(nterms)*(nterms+1) ; Subscripts of diagonal elements
Weights=1/err_y^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
alpha_inv=invert(alpha)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
```

```
par_shell_r(*,i)=a
sig_shell_r_all(*,i)=sigma
Weights=1/err_y_data^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
sig_shell_r_data(*,i)=sigma
Weights=1/err_y_geom^2
alpha = transpose(pder) # (Weights # (fltarr(nterms)+1.)*pder)
sigma=sqrt( 1/alpha[diag] )
alpha_inv=invert(alpha)
sigma = sqrt( alpha_inv[diag] ) <10*sigma
sig_shell_r_geom(*,i)=sigma

endif else begin
  yfit=dblarr(n_el)
  par_shell_r(*,i)=[0.0, lamb_ideal, tmin, 0.0]

endelse
;plotting data
plot,x,intense(*,i)/l_shell_r(i,first_shell_r(i)), $
       title='Chord '+stremo(i+1)+' '+stremo(r_chord(i)), $
       xra=[x(n_el/4),x(n_el*3/4)]
oplot,x,(intense(*,i)+err_intense)/l_shell_r(i,first_shell_r(i)), $
      linestyle=1
oplot,x,(intense(*,i)-err_intense)/l_shell_r(i,first_shell_r(i)), $
      linestyle=1
oplot,x,yfit+y_sub/l_shell_r(i,first_shell_r(i)),color=4,linestyle=2
oplot,x,y_sub/l_shell_r(i,first_shell_r(i)),color=3
```

```
   oplot,x,(y_sub+err_y_sub)/l_shell_r(i,first_shell_r(i)),color=3, $
         linestyle=2
   oplot,x,(y_sub-err_y_sub)/l_shell_r(i,first_shell_r(i)),color=3, $
         linestyle=2
   oplot,x,y,co=5
   oplot,x,y+err_y,co=5,linestyle=2
   oplot,x,y-err_y,co=5,linestyle=2
   oplot,x,yfit,co=6,linestyle=1
   vline,lamb_ideal,color=2
   if (i eq 9) or (i eq 19) then begin
     xyouts,.85,.94,'Experimental spectra',/normal,charsize=.5
     xyouts,.85,.92,'Fitted spectra',/normal,color=4,charsize=.5
     xyouts,.85,.9,'Ideal wavelength',/normal,color=2,charsize=.5
   endif
   yfit_r(*,i)=y_sub+yfit*l_shell_r(i,first_shell_r(i))
endfor
;calculating the parameters and errors for the parameters

sig_shell_l=sqrt(sig_shell_l_data^2 + sig_shell_l_geom^2)
sig_shell_r=sqrt(sig_shell_r_data^2 + sig_shell_r_geom^2)

amp_shell_l=transpose(par_shell_l(0,*))*sqrt(2*!pi)
sig_amp_shell_l=transpose(sig_shell_l_all(0,*))*sqrt(2*!pi)
offset_shell_l=transpose(par_shell_l(3,*))
sig_offset_shell_l=abs(transpose(sig_shell_l_all(3,*)))

vel_shell_l=(lamb_ideal-transpose(par_shell_l(1,*)))*c/ $
        (lamb_ideal*cos(35*!pi/180))
sig_vel_shell_l= transpose(sig_shell_l_all(1,*))*c/ $
                 (lamb_ideal*cos(35*!pi/180))
```

```
del_lamb=sqrt((transpose(par_shell_l(2,*))^2)>tmin^2)
temp_shell_l=alpha_temp*(del_lamb/lamb_ideal)^2
sig_temp_shell_l=2*alpha_temp*del_lamb*transpose(sig_shell_l_all(2,*))/ $
                    lamb_ideal^2

amp_shell_r=transpose(par_shell_r(0,*))*sqrt(2*!pi)
sig_amp_shell_r=transpose(sig_shell_r_all(0,*))*sqrt(2*!pi)
offset_shell_r=transpose(par_shell_r(3,*))
sig_offset_shell_r=abs(transpose(sig_shell_r_all(3,*)))

vel_shell_r=(lamb_ideal-transpose(par_shell_r(1,*)))*c/ $
        (lamb_ideal*cos(35*!pi/180))
sig_vel_shell_r= transpose(sig_shell_r_all(1,*))*c/ $
                 (lamb_ideal*cos(35*!pi/180))

del_lamb=sqrt((transpose(par_shell_r(2,*))^2)>tmin^2)
temp_shell_r=alpha_temp*(del_lamb/lamb_ideal)^2
sig_temp_shell_r=2*alpha_temp*del_lamb*transpose(sig_shell_r_all(2,*))/ $
                    lamb_ideal^2

end
```

shell_length_mid.pro

```
pro shell_length_mid,center=center,d_r=d_r,shell_max=shell_max, $
              r_chord=r_chord,err_center=err_center,err_rmax=err_rmax, $
              cen_per=cen_per,err_cen_per=err_cen_per, $
              r_shell_l=r_shell_l,n_shell_l=n_shell_l,$
              l_shell_l=l_shell_l,err_l_shell_l=err_l_shell_l, $
              shell_all_l=shell_all_l,err_shell_all_l=err_shell_all_l, $
              r_shell_r=r_shell_r,n_shell_r=n_shell_r,$
              l_shell_r=l_shell_r,err_l_shell_r=err_l_shell_r, $
              shell_all_r=shell_all_r,err_shell_all_r=err_shell_all_r, $
              err_cen_l_shell_l=err_cen_l_shell_l, $
              err_cen_l_shell_r=err_cen_l_shell_r, $
              err_max_l_shell_l=err_max_l_shell_l, $
              err_max_l_shell_r=err_max_l_shell_r, $
              err_per_l_shell_l=err_per_l_shell_l, $
              err_per_l_shell_r=err_per_l_shell_r

default,d_r,1.24 ;distance between the chords
default,center,10.5 ;center location of the plasma in chord units
default,err_center,.25 ;error in the center location (chord units)
;shell_max ;maximun shell radius
default,err_rmax,1. ;error in the maximun shell radius (chord units)
default,cen_per,1 ;location of chord i in shell i (from 0 to 1)
;1.0 puts chord r of shell i-1
default,err_cen_per,.25 ;error in the location of chord i in shell i

;calculating the radius of the chords and shells
r_chord=(dindgen(20)-center+1)*d_r
;finding indicices to use to the radius of the shells
```

```
n_l=where(r_chord le 0)
n_r=where(r_chord ge 0)
;calculating the radius of each shell
r_shell_l=-r_chord(n_l)+cen_per*d_r
r_shell_r= r_chord(n_r)+cen_per*d_r
default,shell_max,max(abs(r_chord))+2*d_r ;maximum shell radius
n_shell_l=n_elements(r_shell_l)
n_shell_r=n_elements(r_shell_r)
r_shell_l(0)=shell_max
r_shell_r(n_shell_r-1)=shell_max

;converting errors to real lengthes
err_center=err_center*d_r
err_rmax=err_rmax*d_r
err_cen_per=err_cen_per*d_r

;initializing length arrays
shell_all_l=dblarr(20,n_shell_l) ;length from shell j to the axis
                                  ;along chord i
err_cen_all_l=dblarr(20,n_shell_l) ;error from the center location
err_max_all_l=dblarr(20,n_shell_l) ;error from the max radius
err_per_all_l=dblarr(20,n_shell_l) ;error from the chord location
shell_all_r=dblarr(20,n_shell_r) ;length from shell j to the axis
                                  ;along chord i
err_cen_all_r=dblarr(20,n_shell_r) ;error from the center location
err_max_all_r=dblarr(20,n_shell_r) ;error from the max radius
err_per_all_r=dblarr(20,n_shell_r) ;error from the chord location

;finding lengths for left side fits
for j=0,n_shell_l-1 do $
```

```
   for i=0,19 do begin
      shell_all_l(i,j)=sqrt(((r_shell_l(j))^2-(r_chord(i))^2)>0.0)
      if shell_all_l(i,j) ne 0. then begin
        err_cen_all_l(i,j)=err_center*(r_shell_l(j)- $
                          abs(r_chord(i)))/shell_all_l(i,j)
        err_per_all_l(i,j)=err_cen_per*(r_shell_l(j))/shell_all_l(i,j)
      endif
   endfor
err_cen_all_l(*,0)=(err_center*abs(r_chord))/shell_all_l(*,0)
err_max_all_l(*,0)=(err_rmax*shell_max)/shell_all_l(*,0)
err_per_all_l(*,0)=dblarr(20)
for j=0,n_shell_r-1 do $
   for i=0,19 do begin
      shell_all_r(i,j)=sqrt(((r_shell_r(j))^2-(r_chord(i))^2)>0.0)
      if shell_all_r(i,j) ne 0. then begin
        err_cen_all_r(i,j)=err_center*(r_shell_r(j)-abs(r_chord(i)))/ $
                          shell_all_r(i,j)
        err_per_all_r(i,j)=err_cen_per*(r_shell_r(j))/shell_all_r(i,j)
      endif
   endfor
err_cen_all_r(*,n_shell_r-1)=(err_center*abs(r_chord))/ $
                             shell_all_r(*,n_shell_r-1)
err_max_all_r(*,n_shell_r-1)=(err_rmax*shell_max)/ $
                             shell_all_r(*,n_shell_r-1)
err_per_all_r(*,n_shell_r-1)=dblarr(20)

;initializing l matrix for left and right fits
l_shell_l=dblarr(20,n_shell_l)
err_l_shell_l=dblarr(20,n_shell_l)
l_shell_r=dblarr(20,n_shell_r)
```

```
err_l_shell_r=dblarr(20,n_shell_r)
;initializing length arrays
l_shell_l=dblarr(20,n_shell_l) ;length from shell j to the axis
                                          ;along chord i
err_cen_l_shell_l=dblarr(20,n_shell_l) ;error from the center location
err_max_l_shell_l=dblarr(20,n_shell_l) ;error from the max radius
err_per_l_shell_l=dblarr(20,n_shell_l) ;error from the chord location
l_shell_r=dblarr(20,n_shell_r) ;length from shell j to the axis
                                          ;along chord i
err_cen_l_shell_r=dblarr(20,n_shell_r) ;error from the center location
err_max_l_shell_r=dblarr(20,n_shell_r) ;error from the max radius
err_per_l_shell_r=dblarr(20,n_shell_r) ;error from the chord location

;calculating l and error in l matrices
for j=0,n_shell_l-2 do begin
   l_shell_l(*,j)=shell_all_l(*,j)-shell_all_l(*,j+1)
   err_cen_l_shell_l(*,j)=abs(err_cen_all_l(*,j)-err_cen_all_l(*,j+1))
   err_max_l_shell_l(*,j)=abs(err_max_all_l(*,j)-err_max_all_l(*,j+1))
   err_per_l_shell_l(*,j)=abs(err_per_all_l(*,j)-err_per_all_l(*,j+1))
endfor
l_shell_l(*,n_shell_l-1)=shell_all_l(*,n_shell_l-1)
err_cen_l_shell_l(*,n_shell_l-1)=err_cen_all_l(*,n_shell_l-1)
err_max_l_shell_l(*,n_shell_l-1)=err_max_all_l(*,n_shell_l-1)
err_per_l_shell_l(*,n_shell_l-1)=err_per_all_l(*,n_shell_l-1)

err_l_shell_l=sqrt(err_cen_l_shell_l^2+err_max_l_shell_l^2+ $
                   err_per_l_shell_l^2)

for j=0,n_shell_r-2 do begin
   l_shell_r(*,j+1)=shell_all_r(*,j+1)-shell_all_r(*,j)
```

```
   err_cen_l_shell_r(*,j+1)=abs(err_cen_all_r(*,j+1)-err_cen_all_r(*,j))
   err_max_l_shell_r(*,j+1)=abs(err_max_all_r(*,j+1)-err_max_all_r(*,j))
   err_per_l_shell_r(*,j+1)=abs(err_per_all_r(*,j+1)-err_per_all_r(*,j))
endfor
l_shell_r(*,0)=shell_all_r(*,0)
err_cen_l_shell_r(*,0)=err_cen_all_r(*,0)
err_max_l_shell_r(*,0)=err_max_all_r(*,0)
err_per_l_shell_r(*,0)=err_per_all_r(*,0)

err_l_shell_r=sqrt(err_cen_l_shell_r^2+err_max_l_shell_r^2+ $
                   err_per_l_shell_r^2)

;converting errors back to chord units
err_center=err_center/d_r
err_rmax=err_rmax/d_r
err_cen_per=err_cen_per/d_r

;plot to testt error calculation
;i=3
;plot,l_shell_l(*,0)
;oplot,l_shell_l(*,0)+err_cen_l_shell_l(*,0),co=2
;oplot,l_shell_l(*,0)+err_max_l_shell_l(*,0),co=3
;oplot,l_shell_l(*,0)+err_per_l_shell_l(*,0),co=4
;oplot,l_shell_l(*,0)+err_l_shell_l(*,0),co=5
;oplot,l_shell_l(*,i),linestyle=2
;oplot,l_shell_l(*,i)+err_cen_l_shell_l(*,i),co=2,linestyle=2
;oplot,l_shell_l(*,i)+err_max_l_shell_l(*,i),co=3,linestyle=2
;oplot,l_shell_l(*,i)+err_per_l_shell_l(*,i),co=4,linestyle=2
;oplot,l_shell_l(*,i)+err_l_shell_l(*,i),co=5,linestyle=2
end
```

VITA

Raymond P. Golingo was born in Auburn, Washington. He was raised on his grandparent's farm during his youth. At the University of Washington, he earned a Bachelor of Science in Aeronautical and Astronautical Engineering in June, 1990. He took a hiatus from school when he joined the Marine Corps Reserve. He returned to the University of Washington and earned a Master of Science in Aeronautics and Astronautics in March 1998. During this time he studied Facial Heat Transfer using Inuit traditions. He has worked on the ZaP project since 1998. In 2002 he won the 2002 Graduate Research Award, from the Pacific Northwest Chapter of the AIAA. His published work is shown below.

Refereed Papers

R.P. Golingo, U. Shumlak, Spatial deconvolution technique to obtain velocity profiles from chord integrated spectra *Review of Scientific Instruments*, **74**(4):2332-7, April 2003

D.J. Den Hartog, R.P. Golingo, Telecentric viewing system for light collection from a z-pinch plasma *Review of Scientific Instruments*, **7**2(4):2224-5, April 2001.

U. Shumlak,R.P. Golingo, B.A. Nelson, D.J. Den Hartog, Evidence of stabilization in the Z-pinch *Physical Review Letters*, **87**(20):205005/1-4, 12 Nov. 2001

U. Shumlak, B.A. Nelson, R.P. Golingo, S.L. Jackson, E.A. Crawford, D.J. Den Hartog, Sheared flow stabilization experiments in the ZaP flow Z pinch *Physics of Plasmas*, **10**(5):1683-90, May 2003

Conference Papers

R. P. Golingo, U. Shumlak, and B. A. Nelson, Formation of a sheared flow z-pinch plasma *AIAA, Aerospace Sciences Meeting and Exhibit, 39th, Reno, NV* AIAA Paper 2001-0798 (AIAA Accession number 16621), Jan. 8-11, 2001

U. Shumlak, B.A. Nelson, R.P. Golingo, D. Tang, E. Crawford, D.J. Den Hartog, D.J. Holly, Preliminary results from the flow-through z-pinch experiment: ZaP *1999 IEEE International Conference on Plasma Science. 26th IEEE International Conference* (Cat.

No.99CH36297):108, 1999

U. Shumlak, B.A. Nelson, R.P. Golingo, D. Tang, E. Crawford, D.J. Den Hartog, D.J. Holly, Theory and design of the flow-through z-pinch experiment: ZaP *1998 IEEE International Conference on Plasma Science* (Cat. No.98CH36221):299, 1998

U. Shumlak, B. Nelson, R. Golingo, S. Jackson, P. Norgaard and K. Yirak, A Flow-Stabilized Z-Pinch Fusion Space Thruster *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit* AIAA Paper 2003-4826, July 20-23 2003