Multi-chord Heterodyne Quadrature Interferometry Measurement of the FuZE Sheared Flow Z - Pinch Plasma

Bonghan Kim

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

Master of Science in Aeronautics and Astronautics

University of Washington

2018

Reading Committee:

Uri Shumlak

Brian A. Nelson

Program Authorized to Offer Degree:
Aeronautics and Astronautics

University of Washington

Abstract

Multi-chord Heterodyne Quadrature Interferometry Measurement
of the FuZE Sheared Flow Z - Pinch Plasma

Bonghan Kim

Chair of the Supervisory Committee:
Uri Shumlak
Aeronautics and Astronautics

The FuZE Z-pinch experiment is a plasma confinement experiment that uses radially sheared plasma flow to stabilize the otherwise unstable Z-pinch plasma. The experiment aims to test sheared flow stabilization in fusion-relevant plasma parameters to investigate sheared Z-pinch confinement's potential for nuclear fusion power generation.The FuZE Diagnostic suite includes He-Ne heterodyne quadrature multi-chord interferometry, which has been used to gain insight into the time evolving density characteristics of FuZE Z-pinch plasma. This thesis covers the experimental setup of the He-Ne interferometry, theoretical background behind heterodyne quadrature interferometry, signal analysis using analog circuitry and a Python routine, and the resulting density measurements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author wishes to express his gratitude to Prof.Uri Shumlak, Prof.Brian Nelson, and Prof.Raymond Golingo for their guidance and give special thanks to Dr.Yue Zhang and Johnathon Barhydt for their work on He-Ne interferometry diagnostics. The author also wishes to acknowledge the combined efforts of all those involved in FuZE and ZaP experiments in past and in present, and the enduring support from his family and friends.

Chapter 1

OVERVIEW OF THE FUSION PLASMA

1.1 Introduction to Fusion Plasma

Nuclear fusion energy holds the potential for a clean, scalable, sustainable, and compact means of generating power. The technological appeal of this possibility is easy to understand. On Earth this new source of energy could greatly contribute to the ongoing efforts to reduce human impact on the environment. In space a nuclear fusion reactor could be a power source suitably compact and powerful for use in high power electric propulsion systems, which are desirable for missions with high ΔV requirements. These promises have driven many in the past decades into the pursuit of nuclear fusion.

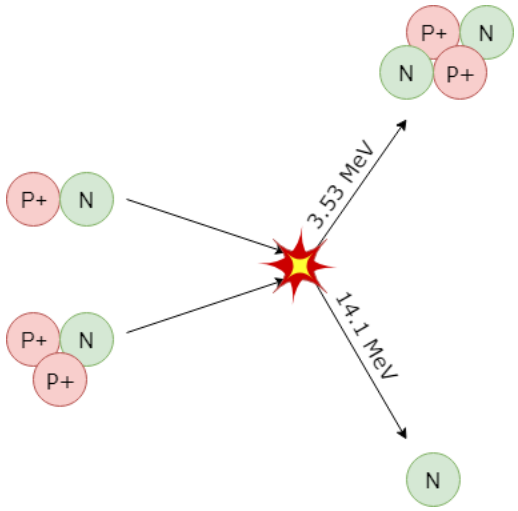

Figure 1.1: Deuterium-tritium fusion reaction yields 14.1 MeV neutron and 3.53 MeV helium. D-T reaction is favored for fusion reactor design due to its high probability for reaction.

Figure 1.1 gives insight into one of the main advantages of nuclear fusion power. The deuterium-tritium (D-T) reaction is one of the most promising choices for a fusion reactor design due to its high probability for reaction. In a D-T fusion reaction each reaction yields a 14.1 MeV neutron as well as a 3.53 MeV helium nucleus. The energy released in a single D-T reaction is far greater than what can be expected from chemical reactions. In comparison the hydrogen-oxygen combustion reaction yields roughly 2.5 eV per reaction. This high energy density makes fusion a keystone technology for high power electric propulsion spacecrafts since the fuel mass is considerably smaller. There are significant amounts of deuterium on Earth, and tritium can be generated as part of the fusion process. Therefore there's very little concern for the eventual depletion of usable fuel. Nuclear fusion produces smaller volume of waste in comparison to nuclear fission, and does not pose significant proliferation concern.

The fusion reaction occurs in conditions of high temperature and density where matter exists in a state of plasma. Understanding of plasma physics is therefore pivotal in efforts toward nuclear fusion. Plasma is often referred to as ionized gas. Ionization processes take place when an atom is energized to a point where the electron escapes the ion. The resulting matter is quasi-neutral and consists of at least two fluids: ion and electron. In a weakly ionized plasma there are also significant neutral atoms present. Plasma readily interacts with electro-magnetic forces due to the charge of the electrons and ions. Plasma is also highly conducting due to the separated charged particles. The nature of plasma gives rise to complex behaviors in addition to that of a typical non-ionized fluid. This interaction between plasma and electromagnetic forces also allows for manipulation of plasma for the purposes of confinement and heating.

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}) \tag{1.1}$$

$$\vec{f} = \rho\vec{E} + \vec{J} \times \vec{B} \tag{1.2}$$

Eq.(1.1) describes the Lorentz force on a charged particle resulting from external electrical

and magnetic field. q denotes the charge of the particle, $\vec{v}$ the velocity of the particle, and $\vec{E}$ and $\vec{B}$ the electric and magnetic fields. The Lorentz force reoccurs in multiple stages of forming a sheared Z-pinch. Eq.(1.1) can also be written in terms of force density $\vec{f}$, charge density ρ, and current density $\vec{J}$ given in Eq.(1.2).

1.2 Z-Pinch Plasma

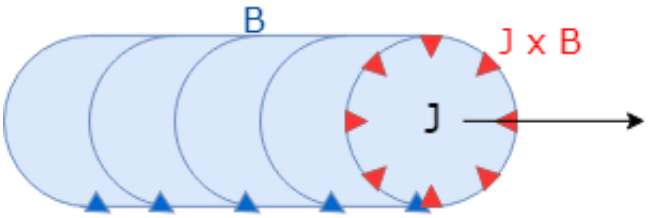

Figure 1.2: Plasma column with axial current and azimuthal magnetic field. The resulting Lorentz force compresses the column radially.

The Z-pinch is one of the early proposed method of plasma confinement and nuclear fusion. A Z-pinch consists of a column of plasma with axial current flowing along the length. The azimuthal magnetic field and the axial current results in a Lorentz force on the plasma that radially compresses the plasma column. Equation 1.3 shows the force equilibrium between plasma pressure and Lorentz force.

$$\frac{dp}{dr} = -J_z B_\theta \tag{1.3}$$

Ampere's law relates magnetic field to current density.

$$\nabla \times \vec{B} = \mu_0 \vec{J} \tag{1.4}$$

For the cylindrically symmetric case the Eq.(1.4) can be expressed as Eq.(1.5)

$$\frac{1}{r}\frac{d}{dr}(rB_\theta) = \mu_0 J_z \tag{1.5}$$

The ideal gas law for plasma is

$$p = n_e k_b \left(T_e + \frac{1}{Z}T_i\right) \tag{1.6}$$

Z is the ionization state $n_e = Zn_i$, k_b is the Boltzmann's constant. Plasma pressure balance can be found using Eq.(1.3), Eq.(1.5), and Eq.(1.6). Assuming that $T = T_i = T_e$.

$$-\frac{d}{dr}\left(p + \frac{B_\theta^2}{2\mu_0}\right) = \frac{B_\theta^2}{\mu_0 r} \tag{1.7}$$

The left-hand side of the Eq.(1.7) includes plasma pressure and magnetic pressure. Balanced by the right-hand side which shows the tension force resulting from curvature of the azimuthal magnetic field.

The Z-pinch has several advantages when compared to other proposed fusion confinement schemes. A Z-pinch does not require external magnetic or electric fields, which eliminates the need for magnetic field coils and the associated technical difficulties. In a Z-pinch, confinement, heating, and compression of plasma is accomplished using one process. This allows for the simplicity of reactor design that is hard to accommodate in other designs. However Z-pinch is not without its own shortcoming — namely its instability.

1.3 Instabilities of the Z-Pinch

The early efforts in Z-pinch fusion faced significant difficulties. The Z-pinch, while simple in concept, proved to be highly unstable. Figures 1.3 and 1.4 shows the zeroth and first azimuthal modes of the Z-pinch instability. In a sausage mode ($m = 0$) instability an initial symmetric radial compression that is irregular about the column's axis results in

concentration of magnetic field around the deformation. This increase in magnetic field leads to further compression. The non-uniformity continues to grow until the current flow is disrupted and Z-pinch dissipates. In a kink mode instability ($m = 1$) an initial kink in a section of a pinch leads to concentration of magnetic fields around the inner radius of the kink, while the outer radius sees more dispersed fields. This leads to force balance that pushes the non-uniform section farther out. Therefore the kink grows until the current flow is disrupted and the pinch is destroyed.

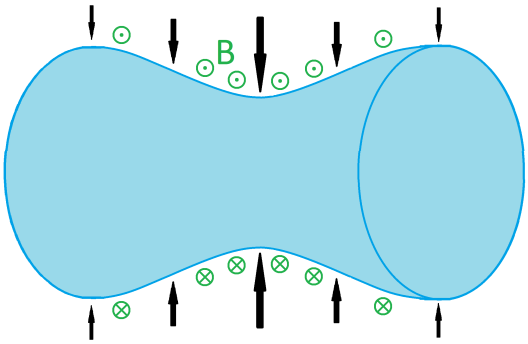

Figure 1.3: Sausage mode instability. The initial radial compression grows as result of positive feedback, until pinch is destroyed.

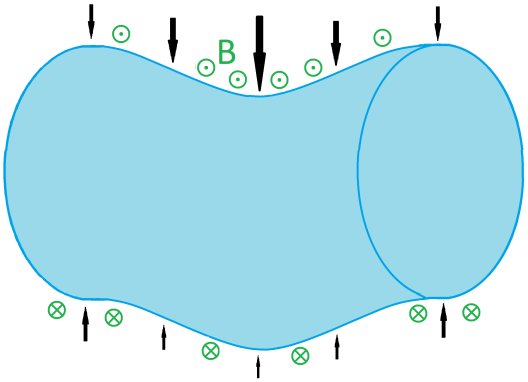

Figure 1.4: Kink mode instability. The initial kink grows as result of positive feedback, until pinch is destroyed.

Several attempts at stabilizing the Z-pinch have been made. Adding a conducting wall near the pinch and imposing external magnetic fields were suggested. A conducting wall near the pinch creates image currents that stabilize the pinch, but requires a conducting wall close to the plasma at 1.2 times the radius of the pinch [8]. This was deemed unfeasible for the purposes of nuclear fusion due to excessive heat flux into the conducting wall. Imposing an external magnetic field required addition of magnetic field coils, which added complexity to the design.

1.4 Sheared Flow Stabilization

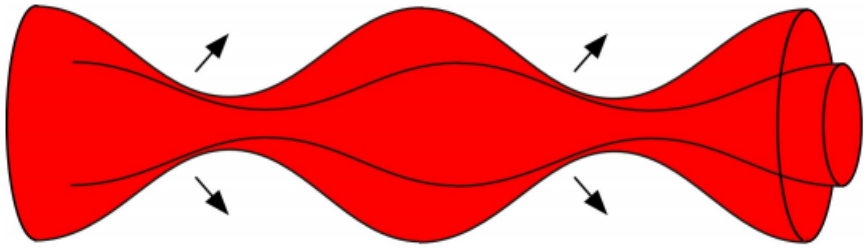

Figure 1.5: Sheared flow stabilization of Z-pinch plasma. Axial flow velocity varies with radius. With sufficient sheared flow Z-pinch can be stabilized. [8]

Sheared flow stabilization stabilizes the Z-pinch using introducing radially varying axial flow velocity. In a sheared Z-pinch the outer layer of the plasma flows at a different axial velocity compared to the inner layer. Any non-uniformity that begins to grow is separated along the radius. Figure 1.6 shows the result from a numerical simulation of sheared flow Z-pinch. The plot shows that without sheared-flow the pinch must rely on a conducting wall for stabilization. This conducting wall can be placed farther away as the sheared flow increases [8].

The simulation result showed stabilization of the m=1 kink mode in a Z-pinch when the radial gradient of the axial velocity meets the condition [8].

$$\frac{dV_z}{dr} \geq 0.1kV_A \tag{1.8}$$

Where k denotes wave number, and V_A is the Alfvén velocity which is

$$V_A = \frac{B}{\sqrt{\mu_0 \rho}} \tag{1.9}$$

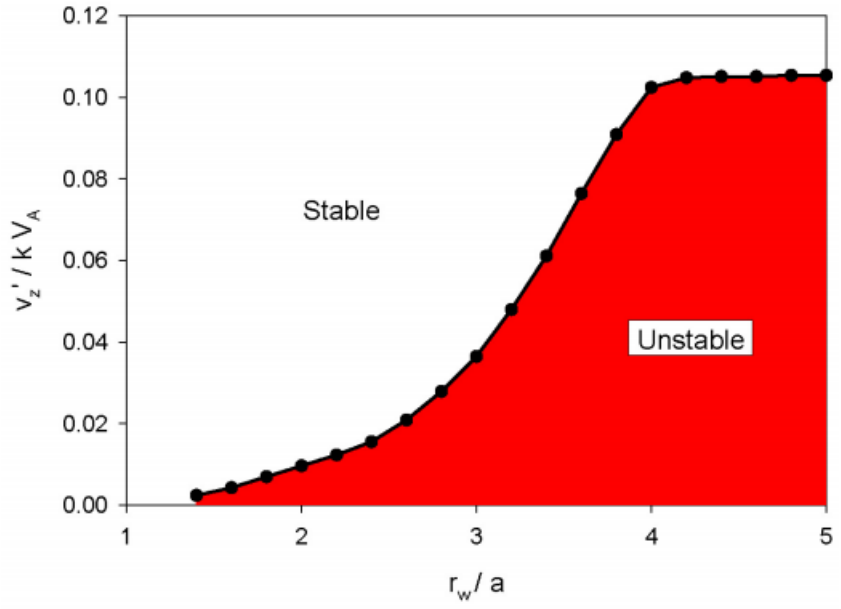

Figure 1.6: The effect of sheared flow on Z-pinch stability. At low sheared flow Z-pinch is only stable with close conducting wall. At sheared flow greater than 0.10 V_z/kV_A Z-pinch is stable without a conducting wall.[8]

The simulation was validated in later experimental works — ZaP and ZaP-HD experiments. The experiments demonstrated stable Z-pinches 100 cm long, 1 cm in radius, exhibiting gross stability for many Alfvén transit times [8]. The success of these experiments provides means to overcome the challenges faced by the early effort in Z-pinch nuclear fusion, and serves as the motivation for the FuZE experiment that seeks to demonstrate sheared flow stabilization in a fusion relevant plasma parameter regimes.

Chapter 2

FUSION Z-PINCH EXPERIMENT

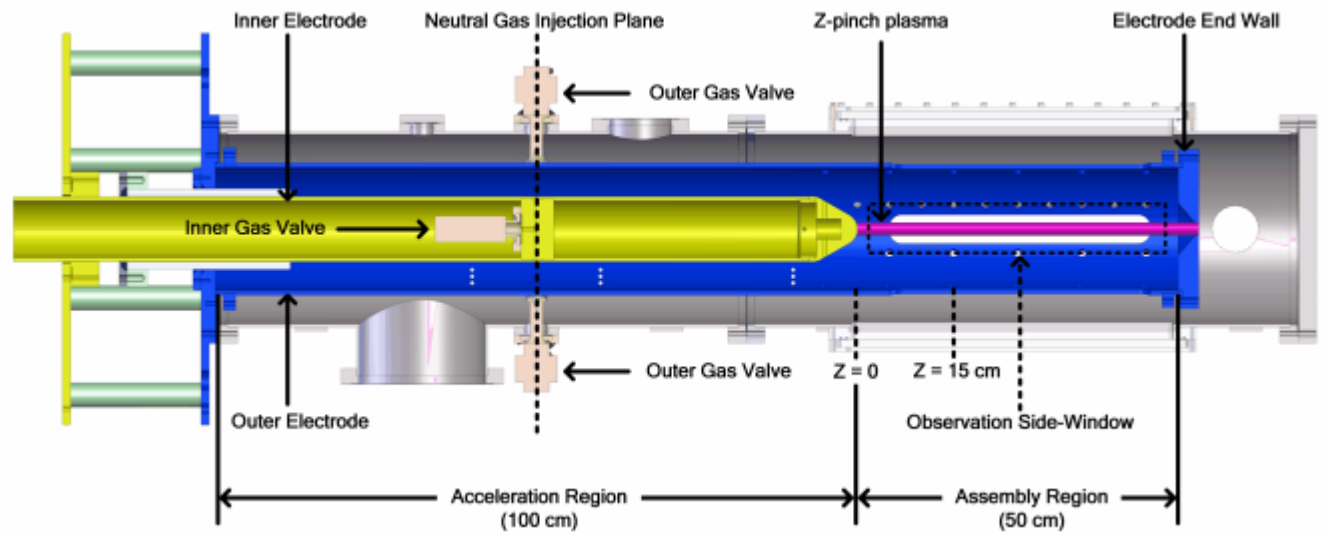

Figure 2.1: Cutaway CAD drawing of FuZE Experiment. Inner electrode is color coded yellow, Outer electrode is color coded blue, and lastly the plasma is color coded purple. [6]

2.1 Design of FuZE Experiment

The demonstrated stability of the sheared flow Z-pinch has motivated further studies, pushing the plasma parameters into fusion-relevant ranges. Latest results from ZaP-HD experiment at University of Washington demonstrated pinches with 0.3 cm radius, 2×10^{17} m^3 plasma electron number density, and 1 keV ion temperature [9], all the while maintaining a stable Z-pinch for the duration of 50 μs. This duration of the pinch is hundreds times greater than the instability growth rate and experimentally confirms the validity of the sheared flow Z-pinch configuration. However, these parameters fell short of the range of fusion-relevant, and

motivated further investigation in form of Fusion Z-pinch Experiment (FuZE). Target plasma parameters in comparison to past University of Washington sheared Z-Pinch experiments are shown below[6].

Table 2.1: FuZE Target Parameters

Parameter	Symbol	ZaP	ZaP-HD	FuZE Target
Capacitor Bank Energy	E_{cap}	144 kJ	50 kJ	250-500 kJ
Charge Voltage	V_c	8 – 10 kV	8 – 9 kV	5 – 10 kV
Plasma Current	I_p	480 kA	250 kA	600 kA
Pinch Current	I_{pinch}	50 – 100 kA	100 – 150 kA	200 – 300 kA
Pinch Length	l_p	0.5 – 1.26 m	0.5 m	0.5 m
Electron Density	n_e	$10^{22} - 10^{23}\mathrm{m}^{-3}$	$10^{23} - 10^{24}\mathrm{m}^{-3}$	$0.1 - 2 \times 10^{24}\mathrm{m}^{-3}$
Temperature	T	75 – 150 eV	600 – 800 eV	600 – 1200 eV
Plasma Lifetime	τ_p	20 – 100 μs	20 – 60 μs	20 – 40 μs
Working Gas		H_2	H_2	H_2, D_2

The operation of the experiment is as follows. The experiment starts with the capacitor banks charged to the selected voltage and the working gas of choice ready to deploy. The timings of capacitor and gas discharge times can be controlled. Initially the gas is discharged from the valves and fill the acceleration region. When the capacitor banks discharge the neutral gas between the inner and the outer electrode ionizes and begins to conduct current. The resulting Lorentz force propels the plasma downstream into the assembly region, where the current connects to the electrode end wall. The resulting axial current flowing through the plasma compresses the plasma column into a Z-pinch.

2.2 Formation of Z-Pinch Plasma

Creation of sheared flow, necessary for the stabilization of the Z-pinch plasma, is central to the design of FuZE experiment. The processes of forming a sheared Z-pinch is shown in Fig. 2.2. (a) Neutral gas is injected via inner and outer gas valves. (b) Capacitor banks discharge, and the resulting voltage across the inner and outer electrode ionizes the neutral gas between the electrodes. Resulting magnetic pressure begins to accelerate the plasma. (c) The plasma pushes out to the assembly region. (d) The plasma makes contact with the outer electrode end wall. The axial current flows through the plasma results in Lorentz forces that compress the plasma column into a Z-pinch. The remainder of the neutral gas that is left in the acceleration region ionizes and accelerates out to the Z-Pinch, replenishing any of the plasma lost out via the end wall and maintaining the sheared flow.

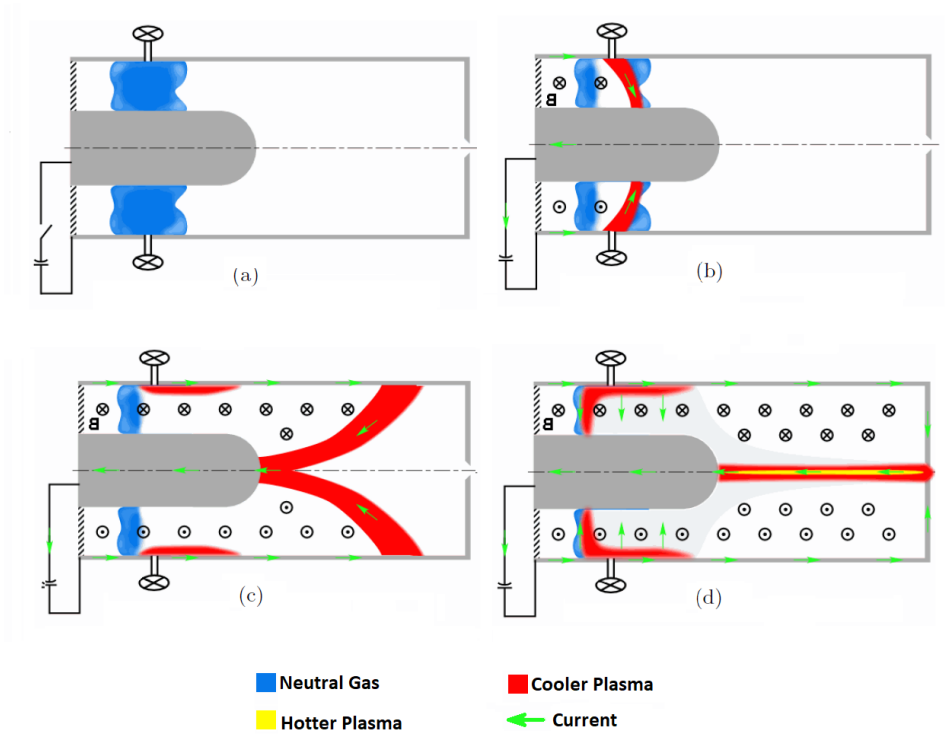

Figure 2.2: Plasma acceleration into the assembly region shown in four stages. Stages are neutral gas injection (a), ionization and acceleration (b - c), and finally compression and Z-pinch formation (d)[2]

2.3 Diagnostic Suite

Gaining insight into the plasma activity during a pulse is a unique challenge. The event lasts 300 μs, of which first 100 μs or so can exhibit quiescent behavior depending on the input parameters. The time resolution required to capture the plasma activity requires a high digitization rate. The highly-energized nature of Z-pinch plasma makes direct measurements unfeasible. To overcome these difficulties a collection of diagnostics are used to establish a more complete understanding of the Z-pinch plasma. Each diagnostic gives information in specific spatial and temporal locations, which can be cross referenced to gain detailed information regarding the Z-pinch plasma.

The magnetic probe array consists of 94 probes arranged along the axial and azimuthal locations. These probes serve as the primary tools to measure the current through the plasma. Azimuthal arrays measure magnetic fields that can not only be used to calculate the total current through the axial location, but it can also give mode information regarding the current centroid location. $m = 0$ mode contain information concerning total current through the axial location, and $m = 1$ mode data, normalized with $m = 0$ data, gives information regarding the x and y current centroid location. When the normalized $m = 1$ mode data is below 0.2 the plasma column is within one radius of the pinch, and is characteristic of a stable Z-pinch [8]. This measurement is frequently used to confirm the stability of the Z-pinch during operation.

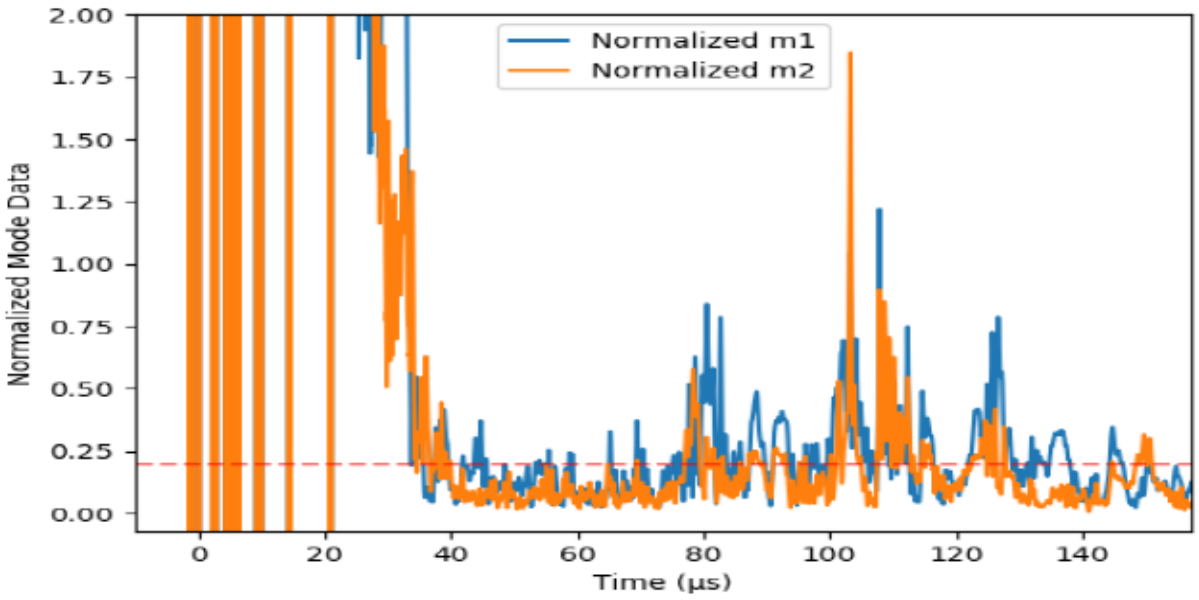

Figure 2.3: $m = 1$ and 2 magnetic probe signals are used to characterize Z-pinch stability. Low mode activity signifies well centered, stable pinch.

Diagnostics that rely on light emission or a laser beam passing through the plasma relies on one of the many view ports present in the vacuum chamber of the experiment. These diagnostics include fast framing imaging camera, time resolved spectrometer, digital holographic interferometry (DHI)[7], and multi-chord He-Ne interferometry. A fast-framing imaging camera is used to directly record the images of the pinch. The spectrometer is capable of measuring light emissions which are used for determining the plasma temperature. DHI provides high spatial resolution plasma density measurement at a single point in time. Neutron detectors measure neutron emission from fusion reactions. Finally He-Ne interferometry measures the temporal evolution of plasma density measurement along the line of passage of a He-Ne laser chord. He-Ne interferometry is the topic of this thesis, but the measurements from other diagnostics are referenced.

Chapter 3

HETERODYNE QUADRATURE INTERFEROMETRY

Interferometry refers to a collection of diagnostic techniques that rely on superposition of waves and resulting interference between the waves to gain information concerning a system. Often times the two waves are from a single source, and the information of the system is embedded into one of the two waves. Interference patterns and knowledge of the wave source allow for extraction of the embedded information. Interferometry relies heavily on the coherence and quality of the wave. A laser is capable of producing a beam of light with high spectral purity and minimal divergence angle. This spatial and temporal coherence of laser light emission makes it a possible to make highly-accurate measurements without directly disrupting the plasma. A laser also interfaces easily with devices such as an acoustic-optical modulator, which are often designed to work with a laser light source. Laser-based interferometry is an important investigative tool in fields ranging from astronomy, remote sensing, mechanical stress/strain analysis, plasma physics and more.

3.1 Introduction to the Interferometry Diagnostic

He-Ne heterodyne quadrature interferometry used in FuZE experiment is a Mach-Zehnder configuration interferometry diagnostic that measures plasma density along the line of passage of a helium neon laser. The measurement of plasma electron density is embedded into phase change of one of the laser beams — a result of variation of the refractive index as a function of electron density.

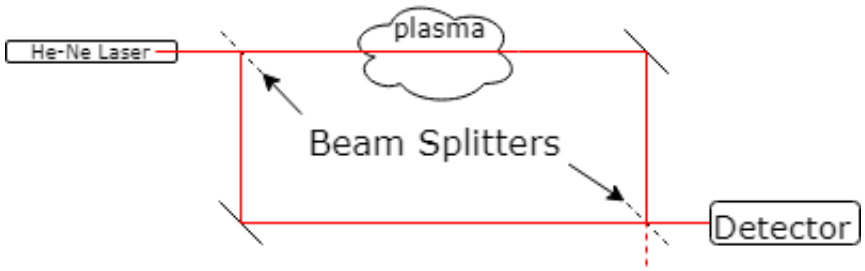

Figure 3.1: Mach–Zehnder configuration interferometry splits a laser beam into two — a scene beam and a reference beam. Scene beam passes through the plasma and results in phase shift that is measured once the beams are combined and measured.

The dispersion relation for a electro-magnetic wave propagating through a cold, magnetized plasma is given by the Appleton-Hartree equation [5].

$$N^2 = 1 - \frac{X(1-X)}{1 - X - \frac{1}{2}Y^2\sin(\theta)^2 \pm ((\frac{1}{2}Y^2\sin(\theta))^2 Y^2\cos(\theta)^2)^{1/2}} \tag{3.1}$$

$$X = \frac{\omega_p^2}{\omega^2}, \quad Y = \frac{\Omega}{\omega}, \quad N = kc/\omega, \quad \Omega = \frac{eB_0}{m_e}, \quad \omega_p = \frac{n_e e^2}{\epsilon_0 m_e^{1/2}} \tag{3.2}$$

N is the refractive index, ω is the wave frequency, ω_p is the plasma frequency, k is the wave number, Ω is the cyclotron frequency, B_0 is the magnetic field, e is the electron charge, m_e is the electron mass, n_e is the charge density, and ϵ_0 is the permissibility of free space.

Equation 3.1 is simplified in case when wave propagation is perpendicular to the magnetic field, and when magnetic field is negligible. In both cases Y $\rightarrow$ 0

$$N^2 = 1 - X = 1 - \frac{\omega_p^2}{\omega^2} \tag{3.3}$$

In a laser beam the total phase lag throughout the length of the beam is given by

$$\phi = \int kdl = \int N\frac{\omega}{c}dl \tag{3.4}$$

However it is difficult to learn about the plasma density via this measurement alone. Firstly, a significant portion of the beam path is outside the plasma, and the reference beam has phase lag that is generally not known precisely. Additional perturbations such as vibrations mean that in order to gain meaningful information the relative phase change should be looked at,

$$\Delta\phi = \int (k_{plasma} - k_0) dl = \int (N-1)\frac{\omega}{c} dl \tag{3.5}$$

Assuming vacuum conditions outside of the plasma $k_0 = \omega/c$. The measurement of the phase shift corresponds to mean refractive index along the line of the beam [5]. By defining $n_c \equiv \omega^2 m\epsilon_0/e^2$ Eq.(3.3) can be written as

$$N^2 = 1 - \frac{\omega_p^2}{\omega^2} = 1 - \frac{n_e}{n_c} \tag{3.6}$$

For small $n_e \ll n_c$ Eq.(3.6) expands to be

$$N \approx 1 - \frac{1}{2}(n_e/n_c) \tag{3.7}$$

Now the Eq.(3.5) simplifies to

$$\Delta\phi = \frac{-\omega}{2cn_c} \int n_e dl \tag{3.8}$$

Resolving the constants

$$\int n_e dl = 5.63 \times 10^{20} \Delta\phi \tag{3.9}$$

In summary the phase shift in the laser beam is proportional to the path-integrated electron number density, as long as the assumptions made are valid. The assumptions are $n_e \ll n_c$, and that the magnetic field term in Appleton-Hartree equation is negligible. Therefore, the

success of the system hinges on accurate measurement of the phase shift throughout the experimental time frame. FuZE He-NE interferometry splits the He-Ne laser into multiple chords, which enables measurements at multiple select locations.

3.2 Heterodyne and Quadrature

In a Mach-Zehnder configuration of interferometry a source beam is split into a scene beam and the reference beam. The scene beam passes through the plasma, while the reference beam bypasses the plasma. Combining these two beams yields a beam with an interference signal. However, measuring density signals from this setup is difficult for two reasons. First, the high laser frequency makes it practically impossible to track the phase shift. Second, there exists ambiguity at phase maxima and minima that makes it impossible to accurately track the density change that produces greater than π phase shift without introduction of quadrature.

The helium - neon laser used in this experiment operates at wavelength of 632.8 nm. This wavelength corresponds to 473.7 THz. This high frequency electro-magnetic wave is difficult to measure directly. In order to track the phase shift a heterodyne technique is used. Heterodyne, also known as frequency conversion, involves two waves of similar frequencies that are mixed. This creates a interference pattern oscillating at the offset frequency. By choosing a specific offset frequency the interference pattern can produce measurable signal that can be tracked. This is accomplished using an acousto-optical modulator.

$$\sin(\omega t + \delta t)\sin(\omega t) = \frac{1}{2}\cos(\delta t) - \frac{1}{2}\cos(2\omega t + \delta t) \tag{3.10}$$

Using a photodetector to measure the intensity results in the high frequency term being averaged away. The resulting signal from the photodetector only includes the oscillation at the offset frequency. While heterodyne allows photodetector to measure the changing beat pattern it is difficult to simply relate this measurement to phase shift. The phase shift signal

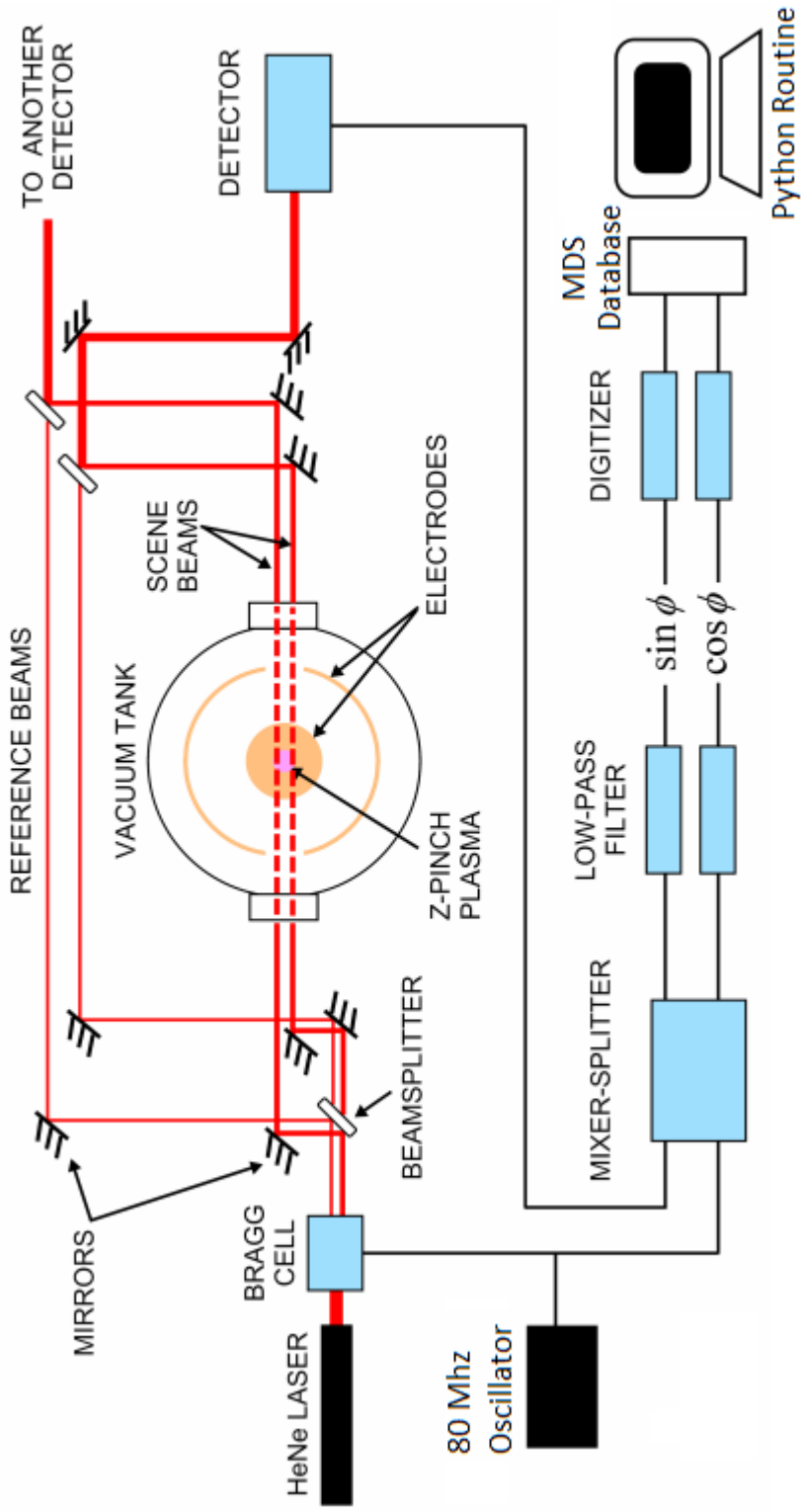

Figure 3.2: The outline of the physical setup of the FuZE He-Ne heterodyne quadrature showing two chords. The setup includes laser, opto-mechancial components, analog circuitry for signal analysis, digitizers, and a computer that processes automated Python routine.

is embedded within the cosine oscillation. Due to the cyclic nature of trigonometric functions this results in discontinuity at fringes of $\pm\frac{\pi}{2}$ when the arctangent function is used to isolate the phase shift signal. At this discontinuity the phase information can not be unambiguously tracked. Any change of θ away from the maximum point of $\cos(\theta)$ will result in decrease in $\cos(\theta)$. At the extremes of trigonometric functions it is not possible to simply recover the input to the function.

In order to overcome this difficulty a quadrature is introduced to the system. In a quadrature system two trigonometric functions (usually sine and cosine signals) are simultaneously used to keep track of the phase shift. This allows for disambiguation of the inverse trigonometric functions because the extremes of two trigonometric functions do not overlap. More detailed step by step mathematical details are included in the next section.

3.3 Signal Analysis: Analog Circuitry

Analog circuitry has a few crucial advantages over digital signal processing that allows for this diagnostic to function. Firstly the analog circuitry acts continuously without discrete digitization at certain times. Secondly the analog circuitry can act at timescales that would be difficult to attain using digital systems. As a result the initial stages of signal handling is accomplished using analog circuitry. Figure 3.3 outlines the electronic circuitry used for data acquisition and initial stage analog electronic analysis.
Once the scene beam has passed through the plasma, and the reference beam has bypassed the experiment, they are recombined into a collimated beam incident on the photodetector.

$$\begin{aligned} E_{scene} &= A_s e^{i(\omega t+\phi(t))} \\ E_{reference} &= A_r e^{i(\omega+\omega_{AOM})t} \end{aligned} \tag{3.11}$$

$\phi(t)$ is the phase shift information embedded into the scene beam, and the ω_{AOM} denotes the chosen acousto-optical modulator frequency. In FuZE experiment an 80 Mhz signal is used. The intensity of the combined beam is measured by the photodetector.

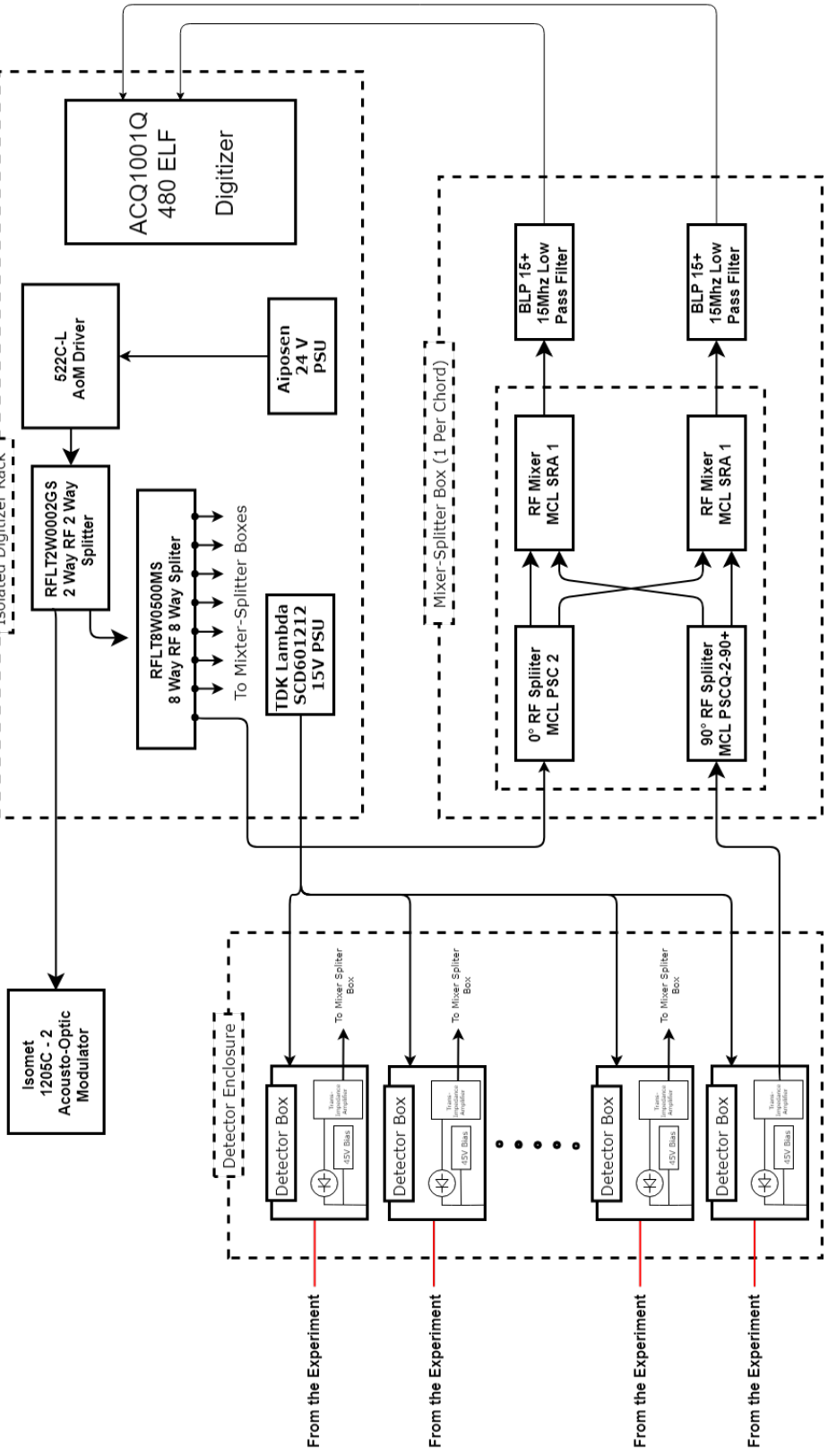

Figure 3.3: Electrical components of FuZE Interferometry System. Major components are grouped by their physical locations and functionality.

$$\begin{aligned} I_{beam} &= (E_s + E_r)^2 \\ &= A_s^2 \cos^2(\omega t + \phi(t)) \\ &\quad + A_r^2 \cos^2(\omega t + \omega_{AOM}) \\ &\quad + 2A_s A_r \cos(\omega t + \phi(t)) \cos(\omega t + \omega_{AOM}) \end{aligned} \tag{3.12}$$

Applying the trigonometric identity $\cos(a)\cos(b) = \cos(a-b) + \cos(a+b)$

$$\begin{aligned} I_{beam} &= A_s^2 \cos^2(\omega t + \phi(t)) \\ &\quad + A_r^2 \cos^2(\omega t + \omega_{AOM}) \\ &\quad + A_s A_r \cos(\omega t + \phi(t)) \cos((\omega + \omega_{AOM})t) \end{aligned} \tag{3.13}$$

The photodetector's response time is much slower than the laser frequency ω, resulting in averaging. $\cos(\omega t)$ terms are averaged to zero, and $\cos(\omega t)^2$ terms are averaged to $1/2A_{s,r}$

$$I_{measured} = \frac{1}{2}(A_s^2 + A_r^2) + A_s A_r \cos(\phi(t) - \omega_{AOM} t) \tag{3.14}$$

The photodetector's output is directed to a transimpedance amplifier that filters out the DC signal and amplifies the signal around ω_{AOM}

$$I_{trans.amp.} = LO = A_s A_r \cos(\phi(t) - \omega_{AOM} t) \tag{3.15}$$

The reference signal comes from the same source that was used to drive the acousto-optical modulator. Using a 90° power splitter, two off-phase reference signals are produced.

$$\begin{aligned} RF_{0^\circ} &= A_{RF} \sin(\omega_{AOM} t) \\ RF_{90^\circ} &= A_{RF} \cos(\omega_{AOM} t) \end{aligned} \tag{3.16}$$

Using a power mixer the two reference signals are mixed with the trans-impedance amplifier to produce sine and cosine signals.

$$\begin{aligned} S_{sin} &= LO \times RF_{0^\circ} \\ &= A_s A_r A_{RF} \sin(\omega_{AOM} t) \cos(\phi(t) - \omega_{AOM} t) \\ S_{cos} &= LO \times RF_{90^\circ} \\ &= A_s A_r A_{RF} \cos(\omega_{AOM} t) \cos(\phi(t) - \omega_{AOM} t) \end{aligned} \tag{3.17}$$

Using the trigonometric identity $2\sin(a)\cos(b) = \sin(a+b)\cos(a-b)$

$$\begin{aligned} S_{sin} &= \frac{1}{2} A_s A_r A_b \sin(\phi(t)) + \cos(\phi(t) - 2\omega_{AOM} t) \\ S_{cos} &= \frac{1}{2} A_s A_r A_b \cos(\phi(t)) + \cos(\phi(t) - 2\omega_{AOM} t) \end{aligned} \tag{3.18}$$

The acousto-optical modulator frequency is filters using 15 MHz low pass filter that effectively isolate the phase shift information.

$$\begin{aligned} S_{sin} &= \frac{1}{2} A_s A_r A_b \sin(\phi(t)) \\ S_{cos} &= \frac{1}{2} A_s A_r A_b \cos(\phi(t)) \end{aligned} \tag{3.19}$$

The resulting signals from the mixer-splitter boxes are digitized in an ACQ1001Q-480 ELF digitizer at a 50 MHz digitization rate. From this point on the signal analysis is done using an automated Python routine. The steps for the python digital signal processing is outlined in next section.

$$\phi(t) = \tan^{-1}\left(\frac{A_s A_r A_b \sin(\phi(t))}{A_s A_r A_b \cos(\phi(t))}\right) \tag{3.20}$$

3.4 Signal Analysis: Python Script

The digitized signal measures the phase information in the interference system. In order to reliably translate this into a path integrated number density signal few more steps are necessary. The outline of the steps are shown in Fig.3.4. The Python code with comments are included in the appendix.

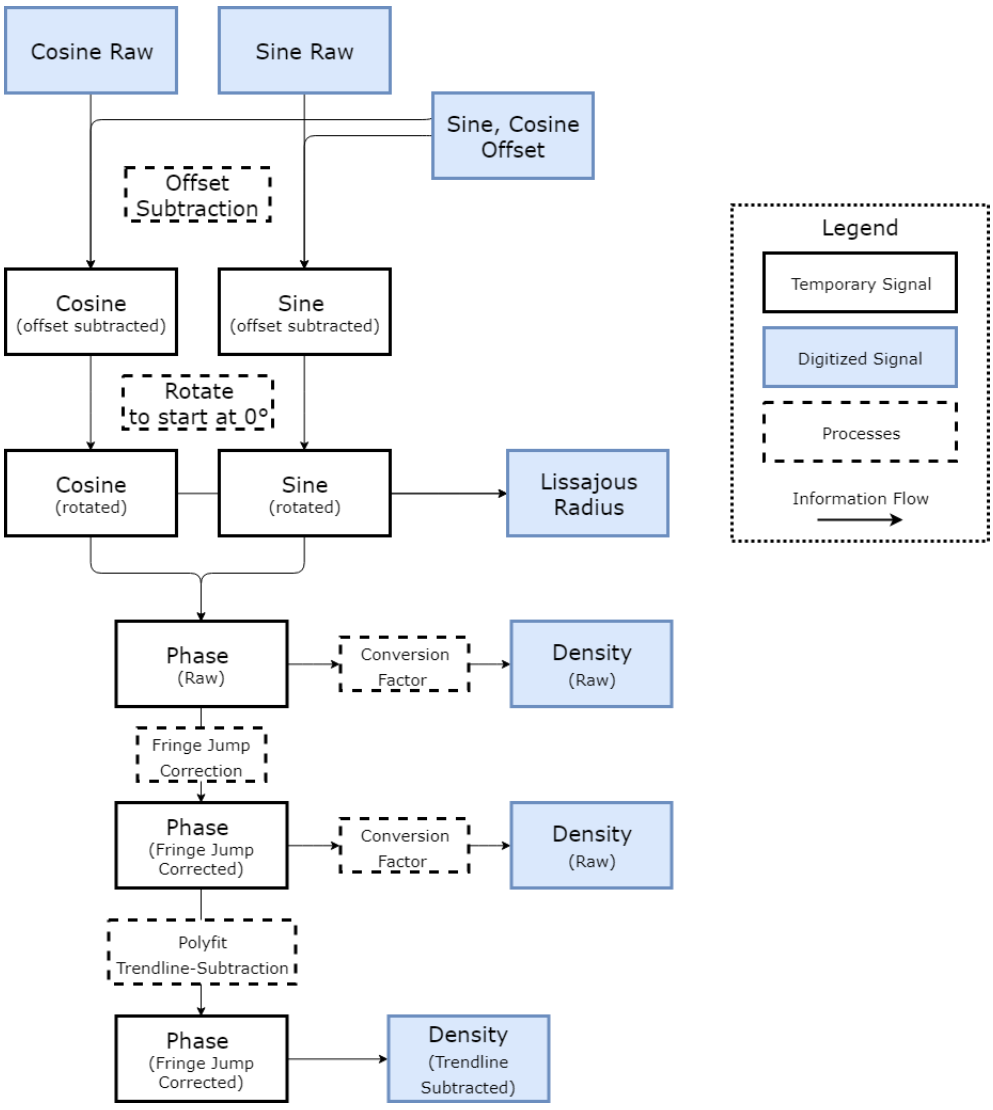

Figure 3.4: FuZE He-Ne IF Python automated signal processing routine. The data are retrieved from and saved to MDS plus database.

Figure 3.5 shows the plot of the digitized signal during plasma pulse, where the cosine signal is plotted on the horizontal axis, and the sine signal is plotted on the vertical axis. The resulting trace draws a Lissajous curve. Due to electrical issues the center of the Lissajou is not always at zero voltage. During the beginning of the day a baseline shot is taken to establish a sine and cosine signal offset. This offset persists throughout the day, but a new baseline can be established if a drift is found during operation.

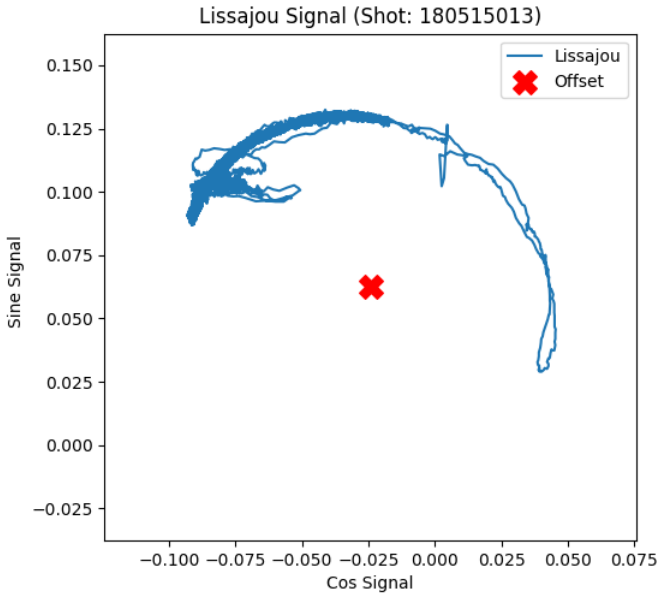

Figure 3.5: Lissajou plot of cosine and sine signals. The phase information manifests in angular rotation of the point on the Lissajou. The magnitude and the consistency of the radius of the Lissajou is indicative of the beam alignment and signal quality.

Rotating the signals such that the initial phase angle is set to zero can improve the signal quality. Not only will this ensure that the signal starts at zero, this will place the initial angle farthest away from the fringe jump. Treating each data point as a coordinate the cosine and sine signals are rotated by the initial angle. The net effect of the offset subtraction and initial angle rotation is shown in Figures 3.6 and 3.7.

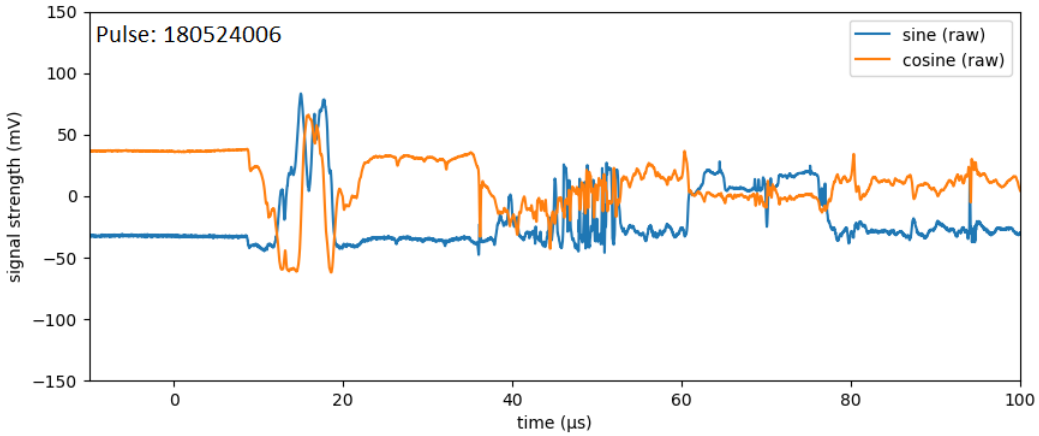

Figure 3.6: Digitized sine and cosine signal before the initial signal processing is applied. The initial angle of the signal is not centered at zero degrees.

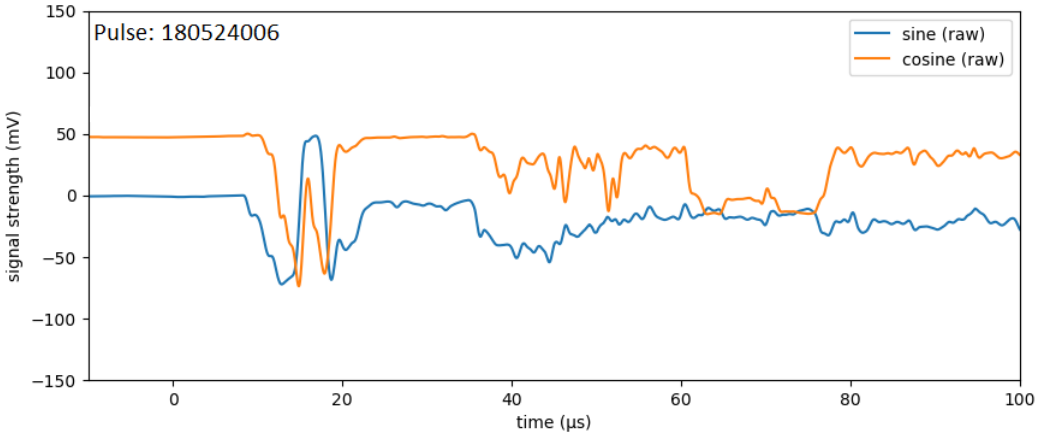

Figure 3.7: Sine and cosine signal after the initial signal processing is applied. The initial angle of the signal is centered at zero degrees.

The raw sine and cosine signals allow for easy conversion to density signal using the arctangent function. Fig. 3.8 shows the raw density signal after the inverse tangent function is applied.

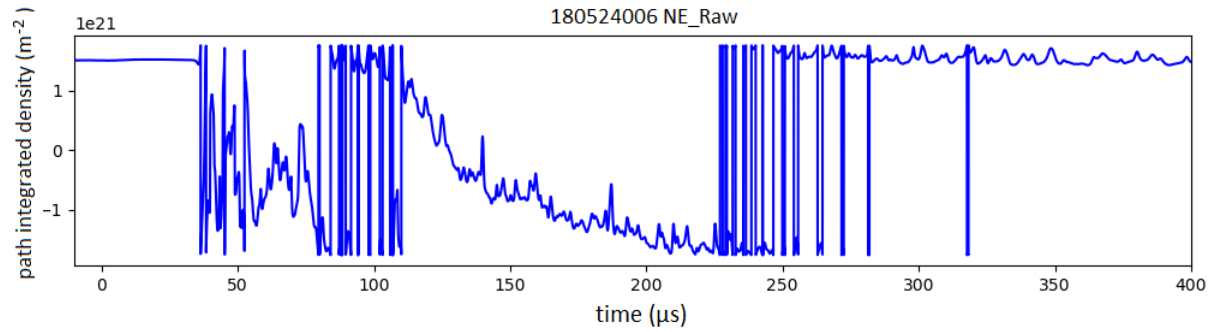

Figure 3.8: Raw ϕ signal contains fringe jumps at $\pm\pi$. The automated routine detects the sudden change in signal, which is characteristic of the phase jump and adds corrections depending on the direction of the phase jump.

Applying the arctangent function to the signal yields the raw density signal. However the discontinuities at $\pm\pi$ resulting from using atan2 function creates a signal with fringe jumps. In order to correct the fringe jump the change of value between data points is used to detect a sudden change in the magnitude close to $\pm 2\pi$. This analysis relies on the assumption that the plasma event does not happen at fast enough time-frame to cause density change of similar magnitude. Applying the phase jump correction results in signal shown in Fig. 3.9

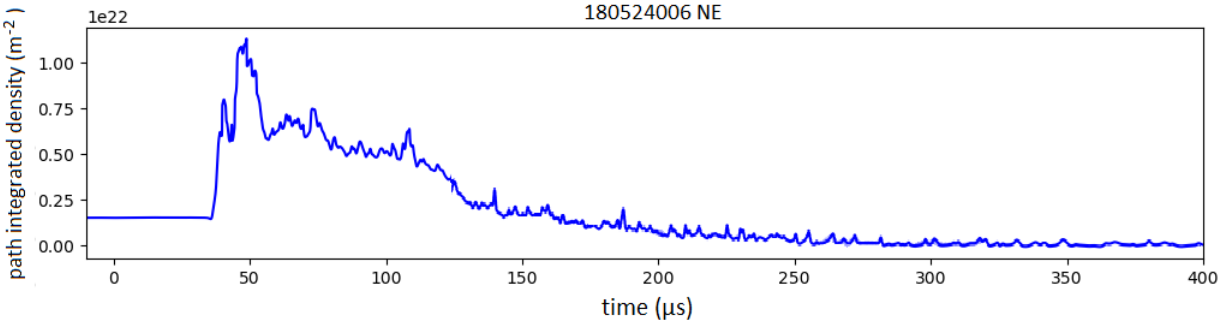

Figure 3.9: Fringe jump corrected signal is now continuous. However there exists a trend line superimposed on the density signal. The source of this trendline is a subject of active investigation. True likely sources are physical vibration and electrical anomalies.

The jump correction removes the abrupt change of signal and provides a continuous signal. However there exists a trendline that is superimposed on top of the signal that results in

discrepancy in the signal before and after the plasma event. Trendline subtraction assumes that the density signal before and after the plasma event are zero. The signal before and after the expected time of plasma event is used to produce polynomial fitting. This polyfit signal is subtracted away to produce trendline subtracted signal. The resulting signal is shown in Fig. 3.10.

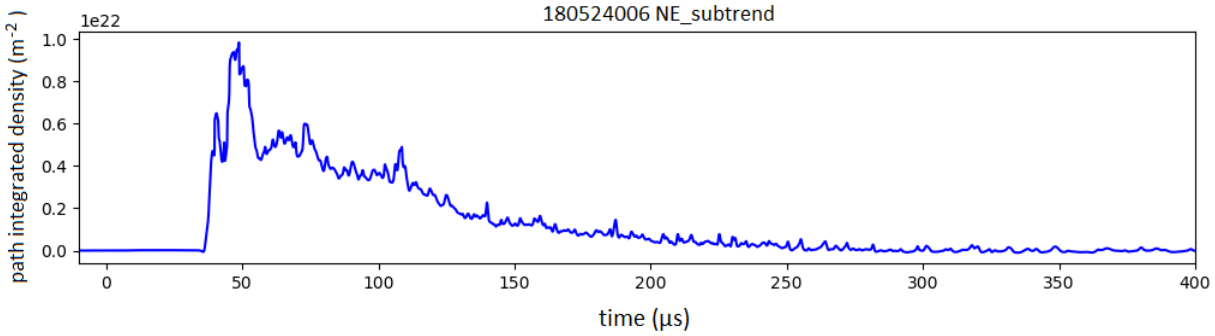

Figure 3.10: Trendline subtraction assumes that the density signal before and after the plasma event are the same. The signal before and after the expected time of plasma event is used to produce polynomial fitting of degree 7. This polyfit function is subtracted away to produce trendline subtracted signal.

This concludes the data processing routine for the FuZE He-Ne heterodyne quadrature interferometry. The diagnostic produces path-integrated density measurement of Z-pinch plasma. The density measurement can then be derived either by assuming a certain pinch radius measured for other diagnostic devices, or by performing an Abel inversion. The results from the plasma diagnostic measurements are covered in next chapter.

Chapter 4

DIAGNOSTIC RESULTS

FuZE operation parameters are numerous. A partial list includes capacitor voltage, inner and outer gas timing, gas pressure, end wall and nose cone configuration. One of the goals of the experiment is to conduct surveys of the parameter regime to find the optimal combination of input parameters that produce the most reliable and scalable Z-pinch. Having an accurate knowledge of the plasma characteristics is crucial. The measurements made using He-Ne interferometry seek to determine the response of the Z-pinch to changing input parameters. This section examines in detail plasma pulses from May 15, 2018 and May 24, 2018.

4.1 Acceleration Region Plasma Density Measurement

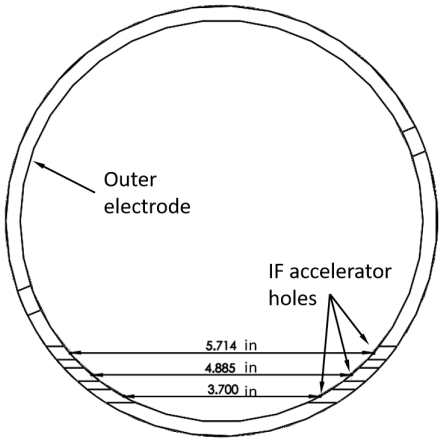

Figure 4.1: Cross-sectional view of the outer electrode showing the locations of the IF accelerator holes.

The behavior of the initial gas injection and acceleration is an important component in understanding the Z-pinch formation and sustainment. The reliability with which the gas valves inject neutral gas, the character of the initial acceleration, and the continued supply of plasma throughout the pinch duration are a few of the important mechanics that need to be understood. In order to study the acceleration of plasma one of the chords targeted the acceleration region at axial location z = -45 cm near injection valves using the top most view-port, shown in Fig. 4.1

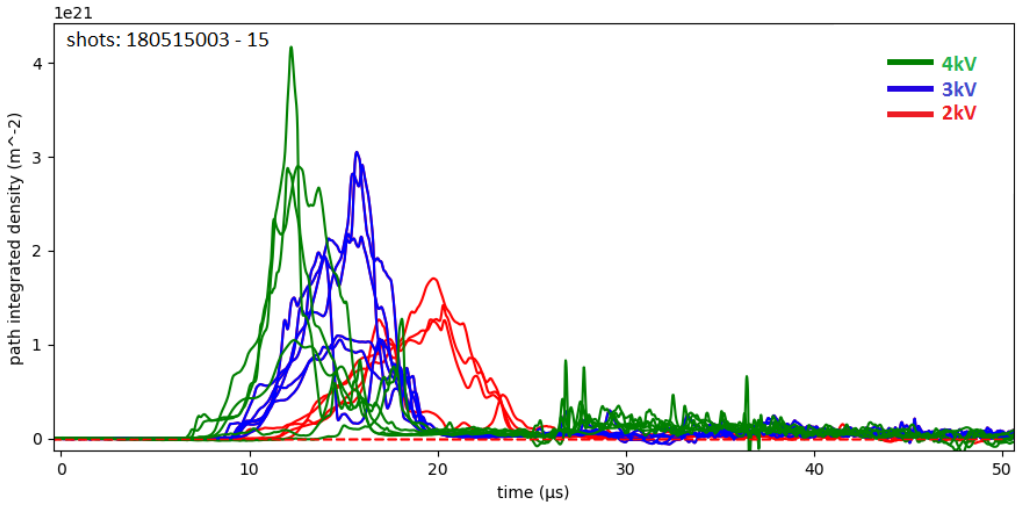

Figure 4.2: He-Ne IF line integrated density measurement at acceleration region z = -45 cm. The signals show initial ionization and acceleration of plasma. Colors denote the capacitor bank voltage. Higher capacitor bank voltage results in shorter duration and more pronounced initial plasma.

The measurements from May 15 in Fig. 4.3 show density measurements recorded in the acceleration region throughout the day. The capacitor bank voltages varied from 2 kV – 4 kV, and the signals are color-coded by the capacitor bank voltage setting. The plot shows fairly consistent behavior within each capacitor voltage group. In all three cases the initial plasma appears 8 μs. The line integrated density peak for higher bank voltages pulses occur earlier in time, with 4 kV pulses peaking at around 11 μs, 3 kV pulses peaking at around 17 μs, and finally 2 kV pulses peaking at around 20 μs. The acceleration plasma also ends sooner

for higher voltage pulses. With 4 kV shots stopping at around 17 μs, 3 kV shots stopping at around 19 μs, and finally the 2 kV shots lasting until 24 μs. These measurements give insight into the characteristic plasma in acceleration region.

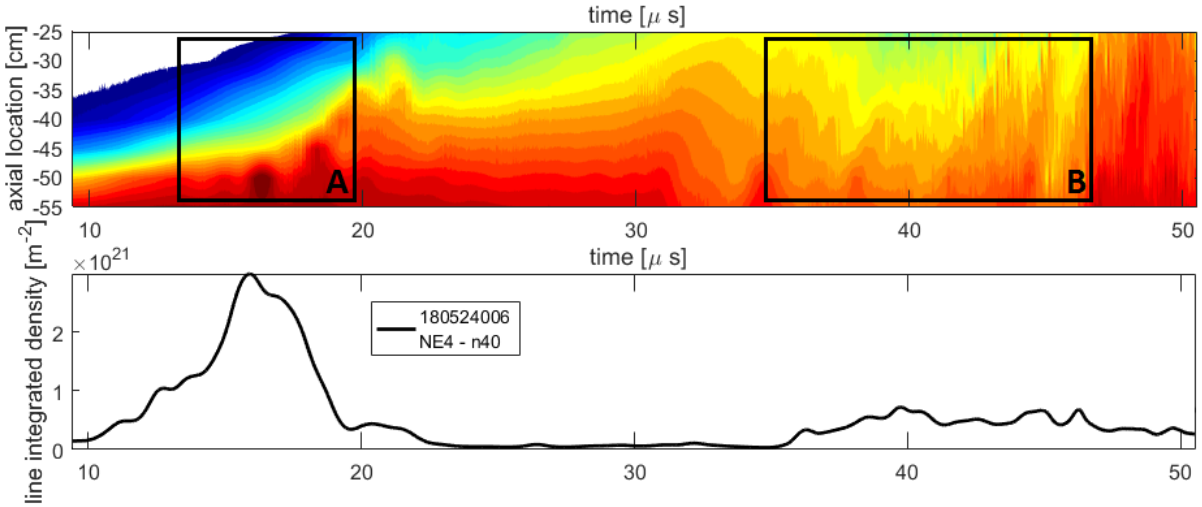

Figure 4.3: (above) Magnetic probe contour plot shows axial current as function of axial location and time. The gradient indicates radial current flow. (below) The He-Ne IF line integrated electron density measurement at acceleration region Z = -45 cm. Two modes of operation — snowplow, and deflagration are marked with A and B respectively.

Magnetic field probe measurements can be cross-referenced for a more in-depth look into the accelerator region operation. Shown in the upper portion of Fig. 4.4 is the magnetic field measurement corresponding to the radial current flow. Shown in box A is the initial acceleration (13 – 20 μs). This mode of operation is referred to as a snowplow mode operation. The working model predicts a forward moving current sheet that ionizes and accelerates the neutral gas in the chamber. This initial push is followed by a relative lull in plasma density measurement (20 – 35 μs). Measured density increases as the magnetic field contour shows forward moving radial current (35 – 50 μs). This mode of operation is referred to as a deflagration mode operation, where the radial current is distributed throughout the acceleration region and results in a continued stream of plasma into the acceleration region. The diagnostics support the idea that once the initial ionized plasma is accelerated out of the

acceleration region, newly ionized plasma begins to flow into the assembly region to replenish the Z-pinch.

4.2 Assembly Region Plasma Density Measurement

The measurement of the time evolution of FuZE Z-pinch plasma is one of the primary goals of the He-Ne interferometer. This is accomplished by placing a chord at locations $z = 5$ – 45 cm, where a pinch is expected.

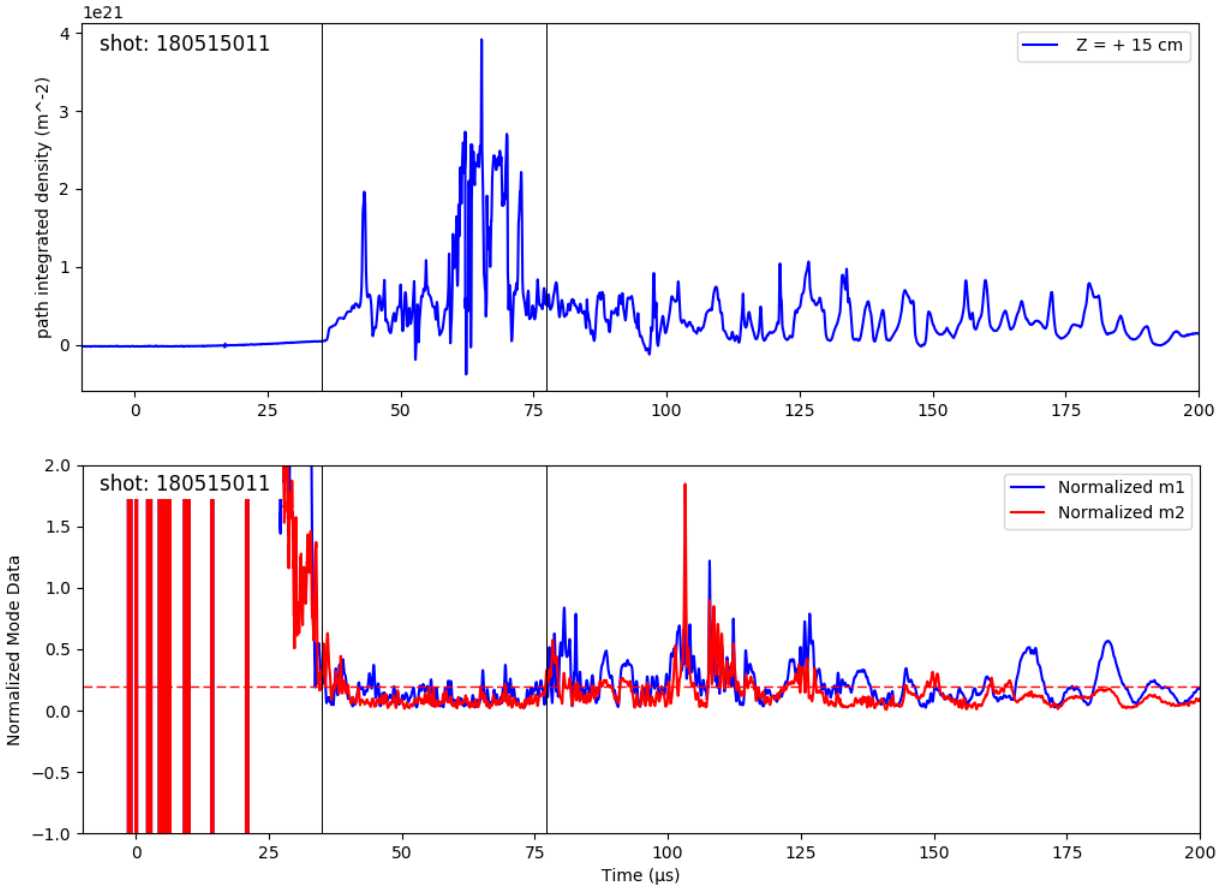

Figure 4.4: (above) He-Ne IF line integrated density measurement at assembly region. (below) Magnetic mode data showing low mode activity during the high line integrated electron density measurement.

The measurement was taken from the pulse 180515011 $z = 15$ cm. The working gas was

helium and the bank voltage was 2 kV. The signal shows plasma first arriving at t = 27.2 μs. There is a local maximum at $t = 28.8$ μs where $1.89\times10^{21}\mathrm{m}^{-2}$ path integrated electron number density is observed. Later during $t = 60 - 75$ μs the path integrated number density measurement shows significantly higher value, peaking at $4\times10^{21}\mathrm{m}^{-2}$. Cross referencing the mode data shows that the quiescent period during the time-frame of the high density. The end of the peak density is followed by the end of the quiescent period.

At higher bank voltage the diagnostic shows higher density measurement. For the shot 180524006 the He-Ne interferometry diagnostic was paired with Digital Holographic Interferometry (DHI) measurement.

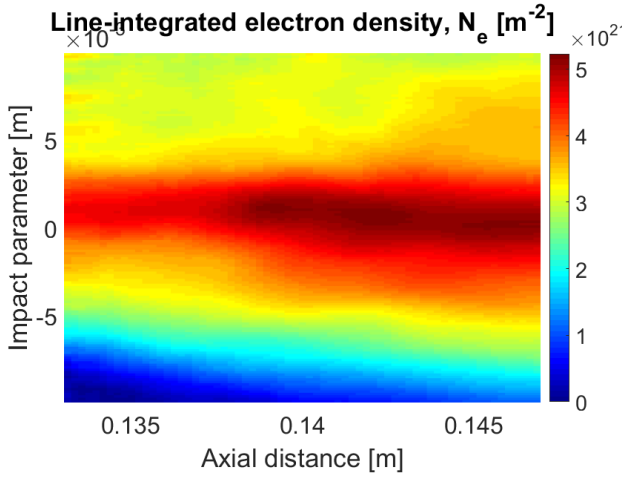

Figure 4.5: Digital Holographic Interferometry Image (raw) shows centered pinch with line integrated electron density up to 5×10^{21}.

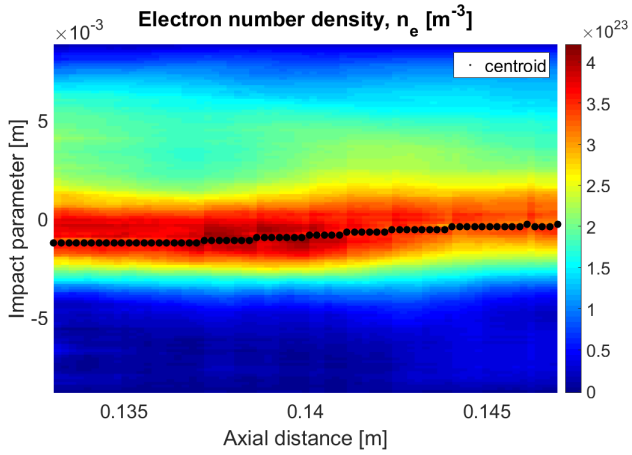

Figure 4.6: Digital Holographic Interferometry image (Abel inverted) shows centered pinch with Abel inverted electron density up to 4×10^{23}.

DHI uses an expanded laser beam to take a holographic image of the plasma at an instant in time, shown in Fig. 4.5. Cross-referencing DHI with He-Ne interferometry is useful due to their complimentary nature — DHI gives highly detailed spatial information at a given time, and He-NE IF gives time-evolving information in a given beam path. The spatial resolution of DHI measurement allows for Abel inversion that deconvolves the holographic image to produce number density as a function of radius, shown in Fig. 4.6. For these images

DHI was targeting $z = 15$ cm where the peak path integrated density shows 4.8×10^{21} m^{-2} path integrated electron number density, which corresponds to 4.1×10^{23} m^{-3} Abel inverted electron number density.

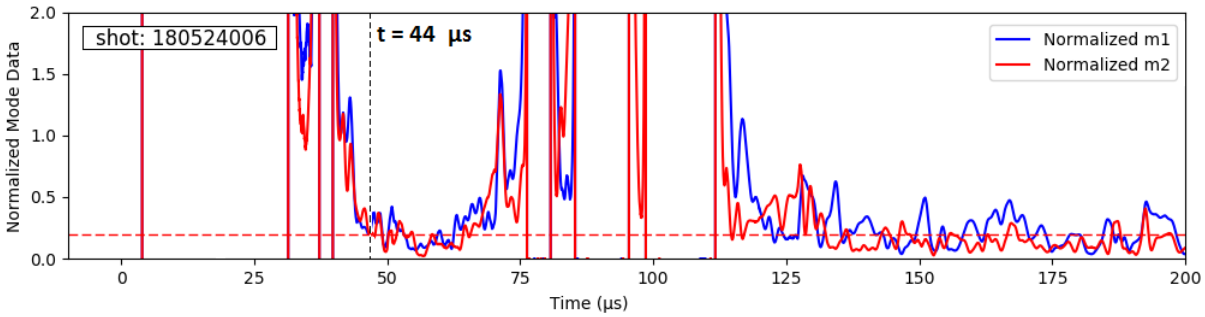

Figure 4.7: Magnetic mode activity (normalized m_1 and m_2) shows low mode activity between 44—60 μs.

The magnetic mode data further supports the existence of a well-behaved Z-pinch during this time. The time between t = 44 μs - 60 μs is characterized with relatively low mode activity.

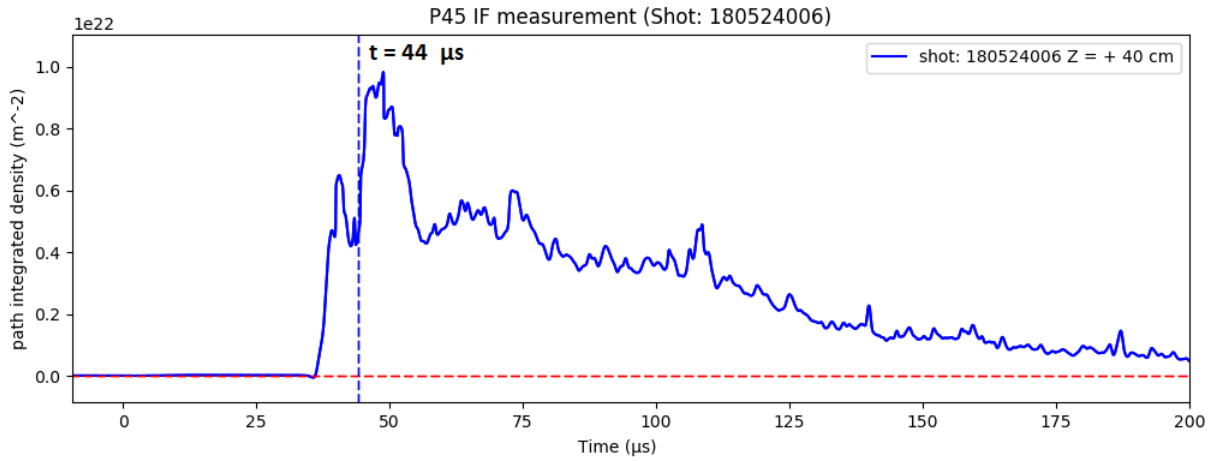

Figure 4.8: Line integrated electron density measurement at acceleration region Z = 40 cm shows He-Ne IF measurement corresponding to the DHI measurement. The measurement is consistent with DHI hologram at 44 μs, and shows peak line integrated density approaching 9×10^{23} m^{-2}

The He-Ne IF signal shows plasma arriving at 34 μs. The signal rises to 4.4 $\times 10^{21}$ m^{-2} path integrated electron number density at the time of DHI measurement, and sharply rises to 9.8 $\times 10^{21}$ m^{-2} at 50 μs. This high density holds until the end of the quiescent period at around 60 μs. The duration of the high density time period is therefore roughly 10 μs. Further more the He-Ne IF path integrated density measurement agrees with the DHI path integrated density hologram. Assuming the pinch radius of 1 cm holds constant during the high density time period the peak number density measures 1.0 $\times 10^{24}$ m^{-3}. This meets the electron density target parameter outlined in Table 2.1.

4.3 Future Work: Abel Inversion

One of the drawbacks of He-Ne IF diagnostic is that the electron density measurements are path integrated. In most cases the measurement of merit is the electron number density per volume. In order to overcome this an Abel inversion can be performed using multiple chords. Unfortunately the electrical noise in FuZE IF system currently poses difficulty in performing a successful Abel inversion. This section will deal with the procedure and feasibility of using Abel inversion in FuZE He-Ne interferometry system in the future.

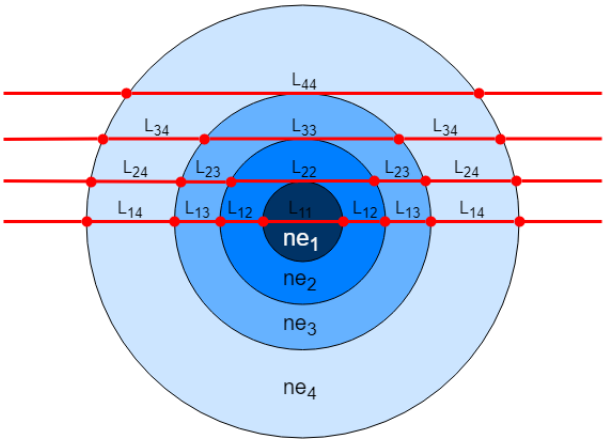

Figure 4.9: Shells of discrete Abel inversion model. The shell boundaries are defined by the laser beam locations. The electron density is assumed to be constant within each shell.

Figure 4.9 shows the discrete shell model used in Abel inversion [4]. In this model the plasma is assumed to be centered and cylindrically symmetric about the plasma axis. This allows the path integrated measurement to be related to radial density profile using following equation

$$F(y) = 2\int_{y}^{\infty} \frac{f(r)r}{\sqrt{r^2 - y^2}} dr \tag{4.1}$$

Discrete shell model allows for piecewise constant radial density profile, where the density of the plasma is constant within the shells defined in radius by the chord locations. With these assumptions the path integrated density measurements $NE_{\#}$ can be related to true number density $ne_{\#}$ using Eq.(4.2). Finding the electron number density is as simple as taking the inverse of the Abel matrix and multiplying the path integrated number density by the inverse matrix.

$$\begin{bmatrix} L11 & L12 & L13 & L13 \\ 0 & L22 & L23 & L23 \\ 0 & 0 & L33 & L34 \\ 0 & 0 & 0 & L44 \end{bmatrix} \begin{bmatrix} ne_1 \\ ne_2 \\ ne_3 \\ ne_4 \end{bmatrix} = \begin{bmatrix} NE_1 \\ NE_2 \\ NE_3 \\ NE_4 \end{bmatrix} \tag{4.2}$$

An alternative model can be made by assuming that the density varies linearly between the shell boundaries. In this case the Abel inversion can be performed by iteratively solving for the number density, with the number density at the shell boundary as the parameter, and the difference between the experimental path integrated density and the expected path integrated density as the error to be minimized.

The number of laser chords passing through the plasma dictates the resolution of the Abel inversion. This is limited by the shape and size of the He-Ne laser beam front, the accuracy of the optics used, and the spacial constriction posed by the opto-mechanical components. As such a synthetic data analysis was performed to test the viability of a good Abel-inverted

measurement in FuZE experimental setting. In this analysis the plasma is assume to have 10 mm diameter, and the beams are placed 5 mm apart from each other.

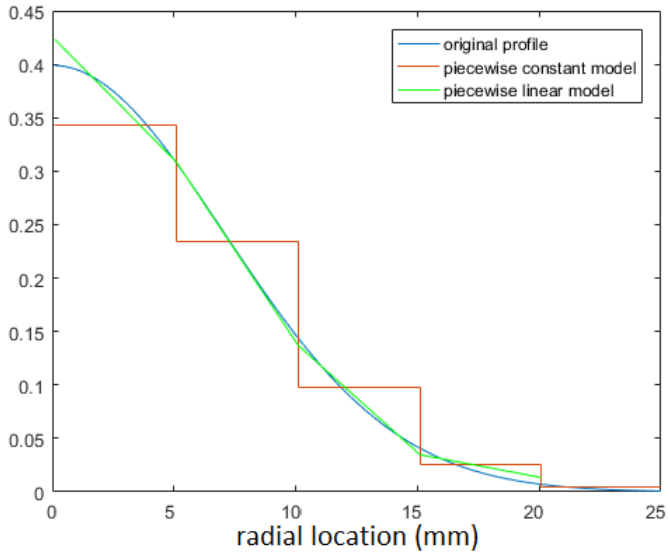

Figure 4.10: Radial Profiles comparing the original synthetic data, Abel inversion using piecewise constant model, and piecewise linear model.

Resulting reconstruction is shown in Fig. 4.10 and in Fig. 4.11. The Abel inversion produces radial and cross-sectional profiles given the chord number and spacing. The error at peak density is 15% for piecewise constant model, and 6.25% for a piecewise linear model. This shows that the current FuZE He-Ne IF system is capable of placing the chord at close enough intervals to perform Abel inversion. However, the assumptions of centered, cylindrically symmetric, Gaussian distributed plasma must be verified before extending this analysis to real life measurements.

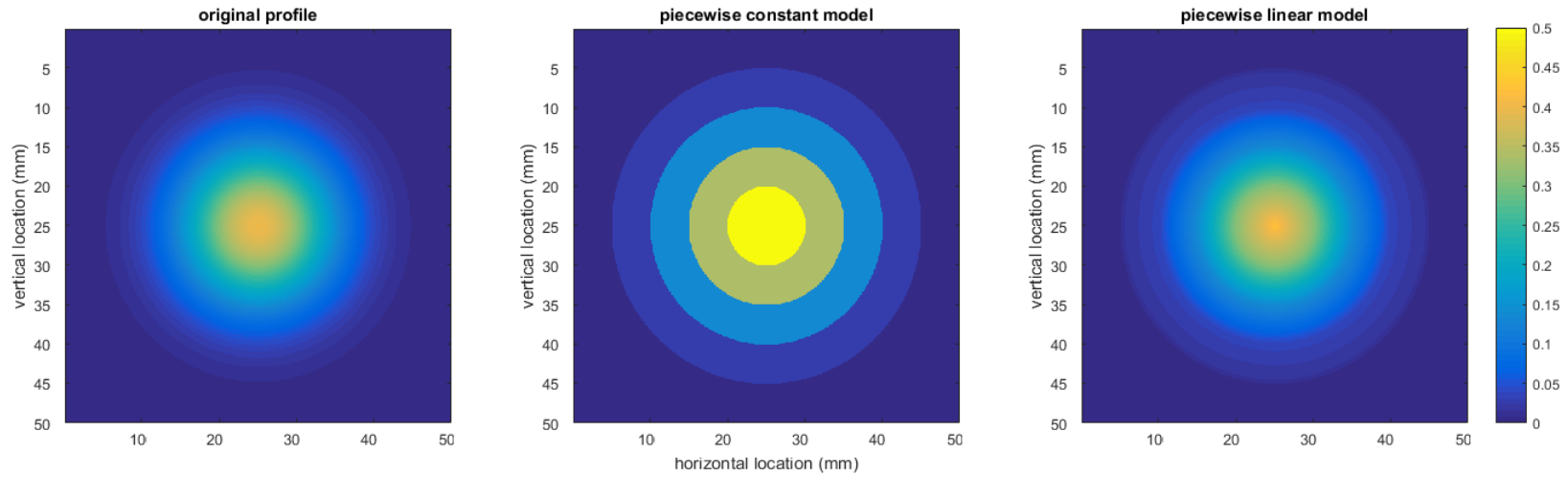

Figure 4.11: Inversion cross-sections comparing the original synthetic data, Abel inversion using piecewise constant model, and piecewise linear model.

Chapter 5

CONCLUSION

The fusion Z-Pinch Experiment (FuZE) is a continuation of the past sheared flow stabilized Z-pinch experiments which includes ZaP and ZaP-HD. Past experiments justified the concept of sheared flow stabilization of Z-pinch. The primary objective of FuZE is to scale the sheared flow stabilized Z-pinch to the fusion-relevant parameter regimes in a step toward demonstrating feasibility of Z-pinch nuclear fusion. A suite of diagnostics are used in conjunction to characterize the Z-pinch plasma, which includes He-Ne heterodyne quadrature interferometry (He-Ne IF). The He-Ne IF diagnostic provides insight into the time evolution of the plasma density by measuring the line-integrated electron density along the beam path. However He-Ne IF alone is insufficient in fully determining the plasma character, since the measurement is limited to the beam paths, and the measurement is line-integrated. In order to overcome this challenge the measurement from He-Ne IF diagnostic is cross-referenced with other diagnostics. Namely Digital Holographic Interferometry (DHI) provides complementary information in the form of a holographic image of the plasma density structure at a given point in time. The He-Ne IF diagnostic was configured to study the initial acceleration of plasma within the acceleration region, and the Z-pinch formation within the acceleration region.

Acceleration region measurements agree with the previously formulated model of accelerator operation. Initial acceleration, referred to as snowplow mode is seen in He-Ne IF measurement as a sharp, high density peak up to 4.1 $\times 10^{21}$ m^{-2} line-integrated electron number density. The initial plasma accelerates out of the acceleration region early in time lasting up to 18 μs for 4 kV pulses, and 25 μs for 2 kV pluses. Later acceleration that replenishes the Z-pinch, referred to as deflagration mode, is seen as a lower density signal measuring roughly

5 $\times 10^{20}$ m^{-2} line-integrated density that persists for longer duration starting around 35 μs ending around 55 μs. The measurement agrees with the magnetic probe measurement, which shows large and forward propagating radial current gradients during the snowplow mode operation, and more distributed and stationary radial current gradient during the deflagration mode operation.

The assembly region measurement agrees with the magnetic probe data and the DHI measurement. On pulse 180524006 the capacitor bank was charged to 5 kV. The peak line-integrated density measured using the DHI was roughly 4.8 $\times 10^{21} m^{-2}$, and at the time He-Ne IF measured 4.4 $\times 10^{21}$ m^{-2}. The peak line-integrated density for He-Ne IF was 9.8 $\times 10^{21}$ m^{-2}. DHI image can be extrapolated using He-Ne IF measurement to obtain the likely electron density during the peak density. In this case the expected electron density is 1.0 $\times 10^{24}$ m^{-3}. This meets the FuZE target parameter for fusion-relevant plasma.

The design of the He-Ne IF system supports up to 8 chords of measurement locations. Future efforts will be on improving the reliability of the system, and introducing additional chord locations. Synthetic data analysis shows that for a 1 cm pinch and 4 chord locations, Abel inversion can yields electron number density within reasonable error range. Additional chord locations at z = -20 cm, and -60 cm could provide more detailed information of the acceleration region operation.

BIBLIOGRAPHY

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

[2] Bei-Jing Chan. Four-chord interferometry measurements of the zap flow z-pinch. Master's thesis, University of Washington, Seattle, Washington, 2008.

[3] I H Hutchinson. Principles of plasma diagnostics: Second edition. *Plasma Physics and Controlled Fusion*, 44(12):2603, 2002.

[4] Sean Knecht. *Comparison of Electrode Configurations on ZaP: Investigation of Heating Mechanisms in a Flow Z-Pinch.* PhD thesis, University of Washington, Seattle, Washington, 2012.

[5] R. Kristal and R. W. Peterson. Bragg cell heterodyne interferometry of fast plasma events. *Review of Scientific Instruments*, 47(11):1357–1359, 1976.

[6] B. A. Nelson, U. Shumlak, E. L. Claveau, E. G. Forbes, R. P. Golingo, A. D. Stepanov, T. R. Weber, Y. Zhang, H. S. McLean, D. P. Higginson, A. E. Schmidt, and K. K. Tummel. Progress on Scaling the Sheared-Flow Stabilized Z-Pinch: The Fusion Z-Pinch Experiment "FuZE". In *APS Meeting Abstracts*, page TO7.001, October 2017.

[7] Michael Ross. *Exploring plasma stability and confinement with high resolution density measurements on the ZaP-HD Flow Z-Pinch.* PhD thesis, University of Washington, Seattle, Washington, 2016.

[8] U. Shumlak and C. W. Hartman. Sheared flow stabilization of the $m = 1$ kink mode in Z pinches. *Phys. Rev. Lett.*, 75:3285–3288, Oct 1995.

[9] U. Shumlak, B. A. Nelson, E. L. Claveau, E. G. Forbes, R. P. Golingo, M. C. Hughes, R. J. Oberto, M. P. Ross, and T. R. Weber. Increasing plasma parameters using sheared flow stabilization of a z-pinch. *Physics of Plasmas*, 24(5):055702, 2017.

[10] 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, 2003.

Appendix A

APPENDIX: PYTHON ROUTINE

Below is the source code of the Python signal processing routine

```
#!/usr/bin/env python
"""
Created on Oct 2017
Last Updated on Dec 13 2017
author: Bonghan Kim
FuZE HeNe interferometry code - Location TODO NOTINPLACE

Version history

V3: Added Brian's optimization and style corrections.
    NOTE: Don't for-loop np.array for simple operations. They can handle
    element wise operations efficiently.

V4: loadSmoothingInfo implemented
  : loadTrendlineInfo implemented
  : Write to tree functions tested

V5: Calculate offset if shotNum == refShotNum
  : Read offset if shotNum != refShotNum
  : Write to offset
```

```
V5 TODO: Write detect IP functionality for detecting the beginning and the end
         of the relevent sig for trendline subtraction.

"""

from scipy.constants import pi
from sys import argv
import MDSplus as MDS
import numpy as np
import matplotlib.pyplot as plt
# Possible packages that I could try and use later
# import pylab as P
# import scipy.signal as signal


def main(shotNum=180214003, chord=1,
         save2tree=False, ignoreZeroR=False, invert=False, debugPlots=True):

    # Get Tree
    T = MDS.Tree('fuze', shotNum)

    # load reference shot number
    refShotNum = T.getNode(r'\NE_1:BASELINE').getData().data()

    if shotNum == refShotNum:
        setRawOffset(T, chord, save2tree, debugPlots)
    else:
        processDensity(T, chord, save2tree, ignoreZeroR, invert, debugPlots)
```

```
def setRawOffset(MDSTree, chord, save2tree, debugPlots):
    (cos_Raw, sin_Raw, time_Raw) = loadIFdata(chord, MDSTree)
    cosOffset = np.mean(cos_Raw)
    sinOffset = np.mean(sin_Raw)

    if save2tree:
        saveDouble(cosOffset, r'\COS_%d:COS_OFF' % chord, MDSTree)
        saveDouble(sinOffset, r'\SIN_%d:SIN_OFF' % chord, MDSTree)

    if debugPlots:
        plt.figure('offset diagnostic plots')
        plt.subplot(2, 2, 1)
        plt.plot(sin_Raw)
        plt.subplot(2, 2, 2)
        plt.plot(cos_Raw)


def processDensity(MDStree, chord, save2tree, ignoreZeroR, invert, debugPlots):

    # ------------------------------------------------------------------------
    # ---------------------------- LOAD SETTINGS -----------------------------
    # ------------------------------------------------------------------------

    convert = 5.61*1E20  # Conversion using 632.8nm for laser wavelength
    T = MDStree
```

```
# load reference shot numbers
refShotNum = T.getNode(r'\NE_1:BASELINE').getData().data()
Tref = MDS.Tree('fuze', refShotNum)

# load smoothing window
(sm_win1, sm_win2, sm_win3, sm_win4) = loadSmoothingInfo(chord, T)

# Threshhold used for detecting fringe jump.
jumpDet = 1.5*pi

# Import IF data from MDS tree with default name T
if invert:
    (sin_Raw, cos_Raw, time_Raw) = loadIFdata(chord, T)
else:
    (cos_Raw, sin_Raw, time_Raw) = loadIFdata(chord, T)

# Import IF baseline data from MDS tree with default name T.
if invert:
    (sinOffset, cosOffset) = loadIFbaseline(chord, Tref)
else:
    (cosOffset, sinOffset) = loadIFbaseline(chord, Tref)

# reads IP data to be used in determining trendline fit domain
(Ip, Tip) = loadIpdata(chord, T)

(sig_i_t, plas_i_t, plas_f_t, sig_f_t) = loadTrendlineInfo(chord, T)

# TODO: make auto detection more reliable
```

```
(plas_i_t_auto, plas_f_t_auto) = find_trendline_domain(Ip, Tip)

if plas_i_t == -99.:
    plas_i_t = plas_i_t_auto
if plas_f_t == -99.:
    plas_f_t = plas_f_t_auto

signal_i = timeToIndex(sig_i_t, time_Raw)
plasma_i = timeToIndex(plas_i_t, time_Raw)
plasma_f = timeToIndex(plas_f_t, time_Raw)
signal_f = timeToIndex(sig_f_t, time_Raw)

# if auto detection fails,
if plasma_i < signal_i:
    plasma_i = signal_i + 200  # must be greater than
if plasma_f > signal_f:
    plasma_f = signal_f - 200

# ------------------------------------------------------------------------
# -------------------------- SIGNAL PROCESSING ---------------------------
# ------------------------------------------------------------------------

# Obtain cosine and sine signal with baseline subtraction.
# This centeres the lisajou.
cos_SubBase = cos_Raw - cosOffset
sin_SubBase = sin_Raw - sinOffset

# Find the initial phase of the cos and sin signal.
```

```
phaseInitial = calc_initial_phase(cos_SubBase, sin_SubBase)

# Rotate the cos and sin signal so the phase starts at 0.
# the give pi+1 above and pi-1 below of phase before phase jump.
(cos_SubBase, sin_SubBase) = rotate2d(cos_SubBase, sin_SubBase,
                                      phaseInitial)

# Smooths cos and sin signal using two windows.
cos_SubBase = smooth(smooth(cos_SubBase, window_len=sm_win1),
                     window_len=sm_win2)
sin_SubBase = smooth(smooth(sin_SubBase, window_len=sm_win1),
                     window_len=sm_win2)

# Convert cos_SubBase and sin_SubBase signals into phase signal with
#  fringe jumps.x
phase = get_phase(cos_SubBase, sin_SubBase)
densityRaw = phase * convert

# Calculates radius of lisajou.
radius = get_radius(cos_SubBase, sin_SubBase)

# When change in value between data+ is greater than threashhold.
# it increments the jump_correction list.
jump_correction = get_jump_correction(phase, radius, jumpDet, ignoreZeroR)

# Subtracting jump_correction*2*pi gives the jump corrected phase.
phase_jump_corrected = phase - jump_correction * 2 * pi
density = phase_jump_corrected * convert
```

```
# Trendline Subtraction.
# NOTE: This trucates the signal from index signal_i ot index signal_f.
(baseline, fitline, fitline_time) = get_trendline(phase_jump_corrected,
                                                  time_Raw, signal_i,
                                                  plasma_i, plasma_f,
                                                  signal_f)
phase_sub_TrendLine = phase_jump_corrected[signal_i:signal_f] - baseline
timeForTrendline = time_Raw[signal_i:signal_f]
densitySubTrend = phase_sub_TrendLine*convert

# Saving smoothed signals
density_sm = smooth(smooth(density, sm_win3), sm_win4)
densitySubTrend_sm = smooth(smooth(densitySubTrend, sm_win3), sm_win4)

# ------------------------------------------------------------------------
# ----------------------------- SAVE TO TREE -----------------------------
# ------------------------------------------------------------------------

# TODO Test this
if save2tree:
    saveData(density, time_Raw, r'\NE_%d' % chord, T)  # NE#
    saveData(density_sm, time_Raw, r'\NE_%d_sm' % chord, T)
    saveData(densityRaw, time_Raw, r'\NE_%d_Raw' % chord, T)  # NE#_Raw
    saveData(densitySubTrend, timeForTrendline,
             r'\NE_%d_SubTrend' % chord, T)  # NE_subTrend
    saveData(densitySubTrend_sm, timeForTrendline,
             r'\NE_%d_SubTrend_sm' % chord, T)  # TODO Too long
```

```
    saveData(cos_SubBase, time_Raw,
             r'\COS%d_rotated' % chord, T)  # processed cos signal
    saveData(sin_SubBase, time_Raw,
             r'\SIN%d_rotated' % chord, T)  # processed sin signal
    saveData(radius, time_Raw,
             r'\NE_%d:radius' % chord, T)  # radius of processed cos,sin
    saveDouble(cosOffset, r'\COS_%d:COS_OFF' % chord, T)  # cos offset used
    saveDouble(sinOffset, r'\SIN_%d:SIN_OFF' % chord, T)  # sin offset used

# ------------------------------------------------------------------------
# ---------------------------- DEBUG PLOTS -------------------------------
# ------------------------------------------------------------------------

# Plots for debugging
if debugPlots:
    plt.close()
    plt.figure("IF Diagnostic Signals shot %d chord %d" % (T.shot, chord),
               figsize=(18, 8))

    # Phase and jump corrected phase
    ax = plt.subplot(2, 3, 1)
    plt.plot(time_Raw, densityRaw, label='Original')
    plt.plot(time_Raw, density, 'r-', label='Jump Corrected')
    plt.legend(loc='lower right')
    plt.grid(True)

    # Radius
    plt.subplot(2, 3, 2, sharex=ax)
```

```
plt.plot(time_Raw, radius, 'g-', label="Radius")

plt.title('Shot: %d Chord: %d' % (T.shot, chord))
plt.legend()
plt.grid(True)
plt.ylim(ymin=0)
# ca = plt.gca()

# Details of trendline subtraction (NOTE: Truncated)
plt.subplot(2, 3, 4, sharex=ax)
plt.plot(time_Raw[signal_i:signal_f],
         phase_jump_corrected[signal_i:signal_f], 'r-',
         label='Phase w/o trendline')
plt.plot(time_Raw[signal_i:signal_f], baseline, 'g-',
         label='Trendline')
plt.plot(fitline_time, fitline, 'b.', label='Fitline Domain')
plt.legend(loc='upper right')
plt.grid(True)

# Trendline subtracted phase signal (NOTE: Truncated)
plt.subplot(2, 3, 5, sharex=ax)
plt.plot(timeForTrendline, densitySubTrend, 'b-',
         label='Trendline Corrected')
plt.plot(timeForTrendline, densitySubTrend_sm, 'r-', lw=2,
         label='Trendline Corrected SM')
plt.legend(loc='upper right')
plt.grid()
```

```
        # Plasma current trace
        plt.subplot(2, 3, 6, sharex=ax)
        plt.plot(Tip, Ip)
        plt.plot(Tip, np.ones(Tip.size)*max(Ip)*0.05, 'r-')


# Imports smoothing windows used in the analysis
# TODO: test
def loadSmoothingInfo(chord, MDStree):
    sm_win = MDStree.getNode(r'\ne_%d:SM_WINDOW' % chord).getData().data()
    return(sm_win[0], sm_win[1], sm_win[2], sm_win[3])


# Imports trendline subtraction domains
# TODO: test
def loadTrendlineInfo(chord, MDStree):
    trnd_win = MDStree.getNode(r'\ne_%d:TREND_WINDOW' % chord).getData().data()
    return(trnd_win[0], trnd_win[1], trnd_win[2], trnd_win[3])


# Imports cosRaw, sinRaw, and time_Raw signal given
#  (chord, refShotnum, MDStree = T)
def loadIFdata(chord, MDStree):
    cosRaw = MDStree.getNode(r'\cos%d' % chord).getData().data()
    sinRaw = MDStree.getNode(r'\sin%d' % chord).getData().data()
    time_Raw = MDStree.getNode(r'\cos%d' % chord).dim_of().data()
    return (cosRaw, sinRaw, time_Raw)
```

```
# Imports IP signal given (chord, MDStree =T)
def loadIpdata(chord, MDStree):
    I_P = MDStree.getNode(r'\I_P').getData().data()
    Tip = MDStree.getNode(r'\I_P').dim_of().data()
    return (I_P, Tip)


# Imports baseline value for cos and sin
def loadIFbaseline(chord, MDStree):
    cosOffset = MDStree.getNode(r'\COS_%d:COS_OFF' % chord).getData().data()
    sinOffset = MDStree.getNode(r'\SIN_%d:SIN_OFF' % chord).getData().data()
    return(cosOffset, sinOffset)


# Calculates initial phase
def calc_initial_phase(cos_SubBase, sin_SubBase):
    # Finding initial phase
    phaseInitial = np.arctan2(sin_SubBase[0], cos_SubBase[0])
    # Rotate the sin and cos signal by - initial phase
    return phaseInitial


# Rotate (xArray,yArray) CW by (angleRad)
def rotate2d(xArray, yArray, angleRad):
    xArray_new = xArray * np.cos(angleRad) + yArray * np.sin(angleRad)
    yArray_new = -xArray * np.sin(angleRad) + yArray * np.cos(angleRad)
    return (xArray_new, yArray_new)
```

```
# Calculates phase signal using (cos_SubBase) and (sin_SubBase)
def get_phase(cos_SubBase, sin_SubBase):
    phase = np.arctan2(cos_SubBase, sin_SubBase)
    return phase


# Calculates radius of lisajou using (cos_SubBase,sin_SubBase)
def get_radius(cos_SubBase, sin_SubBase):
    radius = np.sqrt(sin_SubBase**2 + cos_SubBase**2)
    return radius


# When change in value between data is greater than threashhold
# it increments the jump_correction list
def get_jump_correction(phase, radius, jump=pi, ignoreZeroR=False):
    length = len(phase)
    total_correction = 0
    jump_correction = np.zeros(length)
    for i in range(1, length):

        if ignoreZeroR and radius[i] < 0.01:
            pass
        elif (phase[i] - phase[i-1]) > jump:
            total_correction += 1
        elif (phase[i] - phase[i-1]) < -jump:
            total_correction -= 1
```

```
        jump_correction[i] = total_correction

    return jump_correction


# smooths the signal x and returns as y.
def smooth(x, window_len=11, window='flat'):
        # Possible Windows
        # ['flat', 'hanning', 'hamming', 'bartlett', 'blackman']:
        s = np.r_[2 * x[0] - x[window_len-1::-1], x, 2 * x[-1] -
                  x[-1:-window_len:-1]]
        if window == 'flat':   # moving average
            w = np.ones(window_len, 'd')
        else:
            w = eval('np.'+window+'(window_len)')
        y = np.convolve(w/w.sum(), s, mode='same')
        return y[window_len:-window_len+1]


# Calculates trendline
def get_trendline(phase, time_Raw, signalStart, plasmaStart, plasmaEnd,
                  signalEnd):

    phase = phase[signalStart:signalEnd]
    time_Raw = time_Raw[signalStart:signalEnd]
    plasmaStart -= signalStart
    plasmaEnd -= signalEnd
```

```
    section1 = smooth(phase[:plasmaStart], window_len=5000)
    section1_time = time_Raw[:plasmaStart]
    section2 = smooth(phase[plasmaEnd:], window_len=5000)
    section2_time = time_Raw[plasmaEnd:]

    fitline = np.concatenate([section1, section2])
    fitline_time = np.concatenate([section1_time, section2_time])

    z = np.polyfit(fitline_time, fitline, 7)
    p = np.poly1d(z)
    trendline = [p(x) for x in time_Raw]

    return (trendline, fitline, fitline_time)


# Finds trendline domain from IP data
def find_trendline_domain(Ip, Tip):
    IpThreash = max(Ip) * 0.1
    plasmaIndex = np.where(Ip > IpThreash)
    plasmaBeginIndex = plasmaIndex[0][0]
    plasmaEndIndex = plasmaIndex[0][-1]
    plasmaBeginTime = Tip[plasmaBeginIndex]
    plasmaEndTime = Tip[plasmaEndIndex]
    return (plasmaBeginTime, plasmaEndTime)


# Saves Data to MDS Tree
```

```
def saveData(sig, time, signame, MDSTree):
    node = MDSTree.getNode(signame)
    ySig = MDS.Float32Array(sig)
    node.putData(MDS.Signal(ySig, None, time))


# for saving single number to Tree
def saveDouble(data, name, MDSTree):
    node = MDSTree.getNode(name)
    node.putData(MDS.Float64(data))


def timeToIndex(time, time_Raw):
    indexs = np.where(time < time_Raw)
    return indexs[0][0]


# Runs main if you run this program (not importing to other modules)
if __name__ == "__main__":

    argv = [1, 180223033,1]

    nargs = len(argv)
    if nargs == 1:
        main()
    elif nargs == 3:
        shotNum = int(argv[1])
        chord = int(argv[2])
```

```
    main(shotNum=shotNum, chord=chord)
else:
    print("Usage:")
    print("HeNe_IF_processDensity shotNum chord")
```