

**ABLATION DYNAMICS AND INSTABILITIES OF METALLIC PLASMAS
GENERATED USING MEGA-AMPERE-SCALE CURRENT DRIVERS**

by

Jacob Christopher Zier

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in the University of Michigan
2011

Doctoral Committee:

Professor Ronald M. Gilgenbach, Co-Chair
Professor Yue Y. Lau, Co-Chair
Professor Brian E. Gilchrist
Associate Professor John E. Foster

© Jacob Christopher Zier
2011

DEDICATION

To my family and friends

ACKNOWLEDGEMENTS

First and foremost I wish to thank my advisors, Professor Gilgenbach and Professor Lau, without whom this work would have never existed. Everything I learned throughout my graduate career I owe to their patience and wisdom. I wish to thank Professor Gilchrist and Professor Foster for setting aside their valuable time to serve on my doctoral committee. Additionally, I wish to thank Dr. Mehlhorn for his guidance throughout my college career, his support of my fellowship through Sandia National Laboratories and the National Physical Science Consortium, and for his support of the MAIZE facility at the University of Michigan.

I want to thank Professor Hammer and Professor Kusse of Cornell University for their exceptional guidance when I was a very young graduate student masquerading as a visiting researcher in their laboratory, as well as Doctors Jon Douglass, Ryan McBride, and Dave Chalenski (now at the University of Michigan) for their hard work and advice during those experimental campaigns. Many of the experimental techniques used within this work I learned from them.

I wish to thank my fellow graduate students, in particular Matthew Gomez, David French, Sonal Patel, and Ian Rittersdorf for the countless hours spent building the MAIZE facility, helping to run these experiments, and working through the MRT theory. I also greatly enjoyed my friendships with numerous students during my tenure at the University of Michigan.

Last but not least, I wish to thank my family who supported and guided me my entire life, especially my parents Stephen and Leanne; my brother Michael; my grandparents Martin, Virginia, and Alice; and Uncle Martin and Aunt Sheryl.

This research was performed with fellowship support from the National Physical Science Consortium and Sandia National Laboratories, from the United States Department of Energy award number DE-SC0002590, the United States Department of Energy through Sandia National Laboratories award numbers 240985 and 768225 to the University of Michigan, by the Stewardship Science Academic Alliance program of the National Nuclear Security Administration under Department of Energy Cooperative Agreement No. DE-FC03-02NA00057, and from National Science Foundation award number PHY 0903340 to the University of Michigan. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Co., for the National Nuclear Security Administration of the United States Department of Energy under Contract No. DE-AC04-94AL85000.

TABLE OF CONTENTS

Dedication	ii
Acknowledgements	iii
List of Figures	vii
List of Tables.....	xv
List of Appendices.....	xvi
Chapter 1 Introduction.....	1
Chapter 2 Azimuthally Correlated Ablation between Z-pinch Wire Cores.....	7
2.1 Small Interwire Gap Experimental Configuration.....	10
2.2 Large Interwire Gap Experimental Configuration.....	12
2.3 Image and Correlation Analysis	13
2.3.1 Wire Core Lineout Extraction Method.....	13
2.3.2 Wire Pair Correlation Method	15
2.4 Experimental Results	16
2.4.1 Small Interwire Gap.....	16
2.4.2 Large Interwire Gap.....	19
2.4.3 Experimental Results Summary.....	20
2.5 Physical Correlation Mechanisms.....	21
2.5.1 Azimuthal Current Shunting.....	22
2.5.2 Magnetic Compression.....	23
Chapter 3 A General Theoretical Treatment of the Magneto-Rayleigh-Taylor Instability in a Planar Geometry	25
3.1 Idealized Planar Foil Load Geometry	26
3.2 Derivation of the Acceleration for an Expansionless Slab.....	27
3.3 Derivation of the Generalized Dispersion Relation	30
3.3.1 Fluid Layer Equilibrium.....	32
3.3.2 Perturbation from Equilibrium and Linearization	32
3.3.3 The Perturbed Interface Equilibrium Relations	34
3.3.4 The Perturbed Kinetic Pressure	36
3.3.5 The Perturbed Magnetic Pressure	38
3.3.6 The Perturbed Displacement	38
3.3.7 Region 1 Pressures.....	39
3.3.8 Region 2 Pressures.....	40
3.3.9 Region 3 Pressures.....	41
3.3.10 The Dispersion Relation.....	41
3.4 Derivation of the Instability Feed-Through Effect	43
3.5 Specialized Cases.....	46
3.5.1 Rayleigh Case: Two Regions of Infinite Depth.....	46
3.5.2 Taylor Case: Slab of Finite Thickness	48
3.5.3 Harris Case: Slab of Finite Thickness Supported by Magnetic Field	49

3.5.4	Experimental Case: Slab of Finite Thickness Surrounded by Magnetic Field...	52
Chapter 4	Magneto-Rayleigh-Taylor Experimental Configuration	55
4.1	The MAIZE Pulsed Power Driver	55
4.1.1	LTD Brick Configuration	57
4.1.2	Magnetic Cores and Premagnetization.....	59
4.1.3	High Voltage Charging System	62
4.1.4	Pressure System	65
4.1.5	LTD Module Trigger System	70
4.1.6	Experimental Control and Data Acquisition Systems.....	73
4.2	Power Flow and Load Hardware	78
4.2.1	Transmission Line	79
4.2.2	Experimental Chamber Configuration	82
4.2.3	Foil Load Cartridge Hardware	85
4.3	Laser Backlighter Diagnostic	89
4.3.1	Front End Optics	89
4.3.2	Switchyard Optics	95
4.3.3	Imaging Optics.....	97
Chapter 5	Experimental Performance Predictions using a Circuit Model Drive Pulse... 101	101
5.1	SPICE Circuit Model	102
5.2	Matched Load Performance.....	105
5.3	Foil Load Performance with the Expansionless Slab Model.....	107
5.3.1	Performance with ± 70 kV Charge Voltage	108
5.3.2	Performance with ± 80 kV Charge Voltage	109
5.3.3	Performance with ± 90 kV Charge Voltage	110
5.3.4	Performance with ± 100 kV Charge Voltage	111
5.4	Foil Load MRT Growth Predictions from the Expansionless Slab Model	112
Chapter 6	Magneto-Rayleigh-Taylor Experimental Results.....	115
6.1	Large Foil Offset Case	115
6.1.1	Initial Characterization	116
6.1.2	Temporal Evolution Observations	119
6.2	Small Foil Offset Case	131
6.2.1	Initial Characterization	131
6.2.2	Temporal Evolution Observations	133
6.3	Quantitative Image Analysis	137
6.3.1	Perpendicular Laser Transmission Profile Lineouts	139
6.3.2	Parallel Laser Transmission Profile Lineouts.....	141
6.3.3	Wavelengths of Transmission Profiles.....	142
6.4	Foil Plasma Expansion	145
6.5	Magneto-Rayleigh-Taylor Instability Growth.....	151
6.5.1	Experimental Measurements of MRT Growth	151
6.5.2	Comparison to Expansionless Slab Model Growth Predictions	153
6.5.3	The Role of Plasma Expansion on Instability Growth.....	158
Chapter 7	Summary and Suggestions for Future Work.....	162
Appendices	168
References	379

LIST OF FIGURES

Figure 2.1	Wire locations and point-projection lines-of-sight for the closely-spaced wire configuration (not to scale). The south X-pinch (green) was timed to image the wire pairs earlier, while the north X-pinch (red) was timed to image the pairs later in time. The laser backlighter beams (not shown) passed through the array from left to right. ⁴⁴	10
Figure 2.2	Experimental configuration for the COBRA-STAR imaging system and wire array load. The Z-pinch (A) was located in the center and was imaged by 5 X-pinches (B) in the current return path onto film packs (C). X-ray sensitive PCD detectors (D) provided timing information. Lead collimation (E) preserved film contrast and plastic supports (F) held up the system anode. ⁴⁶	12
Figure 2.3	An illustration of the lineout extraction algorithm geometry.	14
Figure 2.4	An enhanced contrast, false-color radiograph from shot 762 showing 1.9 mm of two 7.4 μm diameter tungsten wires spaced 240 μm apart at 93 ns and 1.13 MA on the cathode side of the fiducial region (yellow rectangle). The corresponding correlated lineouts are shown below. ⁴⁴	17
Figure 2.5	(a) A front-lit preshot image from laser path 3 from shot 762. (b) The data image at 92 ns from shot 762 showing the coronal plasma around the wire pairs, as well as the local precursor column displaced toward the axis from the wire plane.	19
Figure 2.6	(a) An enhanced contrast radiograph from shot 800, taken using the COBRA-STAR imaging system of ten tungsten wires with 7.4 μm diameters at 100 ns and 1.0 MA. The monofilament obscured one of the wires. Lineouts were extracted from the dotted regions within the false color portion of the image. (b) The corresponding lineouts from wires in the radiograph at $6.37 \pm 2\%$ local magnification. ⁴⁴	20
Figure 2.7	Correlation coefficient versus wire separation with 95% confidence intervals for various wire diameters and array diameters in the spatial domain (left) and in the spatial frequency domain (right) with the DC component excluded. ⁴⁴	21
Figure 2.8	A current shunting mechanism for correlated ablation of two neighboring wires (a) prior to gap formation, and (b) after a break formed in left wire. ⁴⁴	22
Figure 2.9	A magnetic compression mechanism for correlated ablation of two neighboring wires in the initial stage (a), and after break had formed in left wire (b). For clarity, only the plasma streamers from the lowest pair of wires were shown. ⁴⁴	23
Figure 3.1	The geometry for the planar foil load region.	26

Figure 3.2	The geometry for the generalized three region MRT instability model.	31
Figure 4.1	(a) An image of the MAIZE LTD with the vacuum and oil flanges, top HV insulator, and top core removed to expose the 40 bricks. KEY: 1) AK gap, 2) Brick subunits (b) The same device with the central vacuum flanges, experimental chamber, and oil section cover installed. KEY: 1) High voltage feed-through and Northstar HV probe. This voltage diagnostic was not available for the MRT shots because the feed-through could not be attached to the cathode hardware without blocking the laser diagnostic, 2) Experimental vacuum chamber bolted to anode vacuum flange, 3) I-beam supports to reduce deflection on the anode flange due to vacuum pressure, 4) Removable oil section cover, 5) Oil section header.	56
Figure 4.2	A cross-sectional cartoon of the brick geometry for a LTD. The insulator slant shown above is for a cathode output electrode on the bottom.....	58
Figure 4.3	A photograph of the premagnetization pulse generator.....	60
Figure 4.4	(a) An image of the top ferromagnetic core during installation. A crane was needed to move the 2 m diameter insulated iron ring into the machine. (b) An image of the top core after being installed into the LTD module. KEY: 1) The top ferromagnetic core installed in the machine.....	62
Figure 4.5	(a) A plot of the approximate LTD core hysteresis loop with the drive pulse (red) assumed to produce a positive field swing, and the premagnetization pulse (blue) producing a negative field swing, resulting in counter-clockwise motion around the hysteresis loop. The relaxation paths for each direction were shown as dashed lines. (b) The same hysteresis loop where the x-axis was expanded to accentuate the loop details around the remnant magnetization points.	62
Figure 4.6	(a) An image of the high voltage supplies used to charge the LTD module. KEY: 1) Positive Glassman high voltage supply, 2) Negative Glassman high voltage supply. (b) The spark gap switch pressure controls, vacuum system controls, and interlock indicator panel for the LTD module. KEY: 1) Pressure controls for trigger switch, 2) Pressure controls for module switches, 3) Letup valve control, 4) Roughing pump valve control, 5) Cryogenic pump gate valve control, 6) Area interlock indicator panel.	63
Figure 4.7	A photograph of the high voltage relay system used to control the MAIZE LTD module. The high voltage connections were placed in a grounded metallic tank filled with insulating oil to prevent arcing. KEY: 1) Positive protection circuits, 2) Positive feed from Glassman high voltage supply, 3) Positive power supply relay, 4) Ground relay, 5) Positive resistive divider voltage monitor, 6) Positive trigger charging feed, 7) Positive LTD module charging feed, 8) Positive dump resistor between capacitors and ground relay, 9) Negative protection circuits, 10) Negative feed from Glassman high voltage supply, 11) Negative dump resistor between capacitors and ground relay, 12) Negative power supply relay, 13) Negative resistive divider voltage monitor, 14) Negative LTD module charging feed.....	64

Figure 4.8	A diagram showing the pressure system for the MAIZE facility's LTD module. KEY: 1) Compressed synthetic air cylinders, 2) LTD module switch system regulator, gauge, and vent, 3) LTD trigger switch system regulator, gauge, and vent, 4) Normally open pneumatic inlet valve for LTD module switches, 5) Normally open pneumatic inlet valve for LTD trigger switch, 6) LTD module switches, 7) LTD trigger switch, 8) 1 PSI check valve, 9) Normally closed pneumatic exhaust valve to vacuum ballast tank and roughing pump (not shown), 10) Shop air compressor, 11) Shop air regulator and gauge, 12) Solenoid valve to LTD chamber letup pneumatic valve (not shown), 13) Solenoid valve to LTD chamber roughing pump pneumatic valve (not shown), 14) Solenoid valve to LTD chamber cryogenic pump gate valve (not shown), 15) Solenoid valve for LTD switch purge cycle.	66
Figure 4.9	An image of the original gas line configuration. The fragile S-shaped gas lines were required due to the alternating gas fitting configuration from switch to switch. KEY: 1) Topside capacitors, 2) Trigger resistors, 3) Spark gap switches, 4) S-shaped gas tubing, 5) Negative charging ring resistors, 6) Positive charging ring resistors.	69
Figure 4.10	(a) An image of the improved gas line configuration for the LTD module spark gap switch pressure system. The switches were connected between markers 1-1 and 2-2. The liquid resistors were also reconstructed using oil-resistant tubing, further improving the reliability of the LTD module and resulting in the green color. (b) The configuration used to crosslink the parallel gas line paths at the inlet and exhaust ports. The switch gas fittings 1-4 were connected to the inlet port via 3 T-shaped connectors.	70
Figure 4.11	The triggering progression for the MAIZE LTD module. KEY: 1) Bi-directional software communication via USB to determine when to trigger the experiment, 2) Software trigger via RS-232 connection, 3) TTL triggers to additional diagnostic systems, 4) 10 μ s TTL gate trigger, 5) +150 VDC bias to capacitance, 6) +150 VDC bias backed by storage capacitance for sufficient current output, 7) ~50 V trigger to thyatron from capacitor discharge, 8) Grounding of intermediate spark gap plane to break down high voltage trigger switch, 9) High voltage trigger to each LTD module quadrant to trigger the machine, 10) Voltage and shorting commands to additional diagnostic systems.	71
Figure 4.12	(a) A side view of the thyatron unit showing the trigger input and the sync output connections. (b) A front view of the thyatron unit. When the circuit became too warm the thyatron did not trigger reliably. The additional cooling fan was added to prevent this heat buildup, which resulted in reliable triggering.	72
Figure 4.13	The high voltage trigger for the LTD module. KEY: 1) Trigger spark gap switch gas line, 2) Cable to thyatron switch, 3) High voltage trigger outputs for the LTD module, 4) Positive charging cable.	73
Figure 4.14	A diagram showing the experimental control and data acquisition components. KEY: 1) TTL <i>Fire</i> and oscilloscope command triggers, 2)	

GPIB data link, 3) Shorting relay for shutters, 4) Monitor for area interlocks, 5) RS232 data link for programming and *Fire* signal, 6) Computer ↔ DAQ USB data link. DAQ→Laser signals were the *Laser Charge* and *Laser Dump* amplifier controls. Laser→DAQ signal was *Laser Monitor* for the amplifier capacitor voltage, 7) DAQ→Relay Control Unit signals were *PS Connect*, *Unabort*, and *Purge Switches*. Relay Control Unit → DAQ signals were *Ground Monitor*, *PS Connect Monitor Pos*, *PS Connect Monitor Neg*, *Area Interlock*, *Equipment Interlock*, and *Ready* signals, 8) TTL trigger to laser Q-switch, 9) RS232 data links for the positive and negative Glassman high voltage supplies, 10) *Voltage Pos* and *Voltage Neg* signals for monitoring LTD capacitor voltages, 11) 120 VAC interface to Ross Relays.....75

Figure 4.15 A screenshot of the Labview control program GUI. The power supplies and interlocks were not set when this particular screenshot was taken, so the power supply status indicated an error.....75

Figure 4.16 (a) An image of the installed DAQ unit’s rear I/O panel showing the 25 pin connector, two screw terminals, numerous BNC connectors, and USB port. KEY: 1) BNC connection ports, 2) Screw terminal 1, 3) Screw terminal 2, 4) USB connector, 5) 25 pin D-sub connector. (b) An image of the DAQ unit’s internal components showing the NI-6229, power supplies, and buffer chip circuitry. KEY: 1) Buffer chip circuitry, 2) 5 VDC supply, 3) 8.3 VDC supply, 4) NI-6229 DAQ, 5) 120 VAC bus.77

Figure 4.17 An image of the relay control unit’s front panel showing the status indicator lights, switch purge button, manual operation controls, abort button, and power key.78

Figure 4.18 A CAD rendering of the MAIZE LTD module MITL sliced along (a) the North-South machine axis, and (b) along the East-West machine axis with the power flow shown in red. KEY: 1) Load cartridge, 2) MITL riser section, 3) MITL transition section, 4) MITL radial section, 5) MITL coaxial section, 6) LTD module AK gap.80

Figure 4.19 (a) A CAD rendering of the MAIZE experimental vacuum chamber with the support beams removed for clarity. The port numbers are labeled in red. (b) An image of the experimental chamber with the laser backlighter path shown in green. KEY: 1) Debris shields, 2) Foil load cartridge and tensioner, 3) Connection shaft for tensioning mechanism, 4) Rotating vacuum feed-through for tensioning mechanism, 5) MITL riser section, 6) B-dot current monitor, 7) Optical fiber.83

Figure 4.20 CAD renderings of the foil load cartridge hardware from the (a) top view, (b) isometric view, (c) front view, and (d) right side view. The anode return current plates are blue, the anode bridge is teal, the tensioner is yellow, and the cathode is red. The foil interaction region is circled in green.86

Figure 4.21 (a) A side view of the load cartridge before loading into the chamber with the support posts installed rigidly connecting the cathode to the anode bridge. The foil was obscured by the support post. KEY: 1) Tensioner with attached foil, 2) Anode bridge, 3) Support posts, 4) Anode current return

	plates, 5) Cathode. (b) An angled view of the load cartridge showing the foil inside the cartridge. Significant slack was left in the foil during transport to reduce the risk of tears. KEY: 1) Foil load. (c) A side view of the load cartridge installed in the experimental chamber with the support posts removed. The foil was only loosely tensioned for the photograph to reduce the risk of tears during the chamber evacuation process. The 1 cm wide interaction region was circled in green. KEY: 1) Cathode MITL riser plate, 2) Anode MITL riser plates, (d) The installed load cartridge showing the tensioning mechanism connected.....	87
Figure 4.22	(a) An image of the Nd:YAG main lasing cavity. KEY: 1) High reflector and cavity length micrometer, 2) Q-switch optical head, 3) Electronic shutter, 4) Nd:YAG laser head enclosure, 5) Brewster window, 6) Mode-locker optical head, 7) Output coupler, 8) Internal photodiode, 9) Manual shutter. (b) An image of the front end beam path. KEY: 1) Pockels cell and crossed polarizers, 2) Amplifier optical head, 3) 50 cm focal length lens, 4) 1.0 neutral density in flip mount, 5) Frequency doubling optical head, 6) 25 cm focal length achromatic doublet lens, 7) 50 μ m diameter aperture spatial filter, 8) 25 cm focal length collimating lens, 9) Removable beam dump, 10) Beam pipe to switchyard.....	90
Figure 4.23	An image of the amplifier storage capacitor and monitoring system. KEY: 1) Voltage divider (\sim 750:1) monitor, 2) R = 550 k Ω bleedoff resistance. RC = 230 s, 3) High voltage electrode attached to Korad supply, 4) Ground electrode, 5) C = 420 μ F storage capacitor.	94
Figure 4.24	An image of the switchyard optical table and beam path. KEY: 1) Alignment iris, 2) 10 cm focal length lens, 3) 50 cm focal length lens, 4) 1% beam sampler, 5) 50/50 beamsplitter, 6) 5 cm focal length lens, 7) Timing photodiode.....	96
Figure 4.25	The geometry for the planar foil load region showing the magnetic fields and laser backlighter path. This optical path measured the y-direction MRT wavelength modes.....	97
Figure 4.25	The imaging optics section for the laser backlighter diagnostic. KEY: 1) 532 nm line filter (1 nm FWHM) and 0.4 neutral density filter for each beam path, 2) 1 m focal length achromatic doublet imaging lenses for each beam path, 3) Camera 1 assembly complete with 0.1 neutral density filter, 532 nm line filter (10 nm FWHM), Canon XSI digital camera, and Faraday shield enclosure, 4) Camera 2 assembly (see #3), 5) Remote trigger cables, (6) Light tight enclosure with lid removed.	99
Figure 5.1	Control level for the assembled LTSPICE model of the LTD module.	103
Figure 5.2	(a) The load current and voltage (x10) performance for \pm 70 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal for \pm 70 kV. (c) The load current and voltage (x10) performance for \pm 80 kV charge voltage. (d) The AK gap insulator stack voltage and capacitor voltage reversal for \pm 80 kV.....	105
Figure 5.3	(a) The load current and voltage (x10) performance for \pm 90 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal for \pm 90 kV. (c) The load current and voltage (x10) performance	

	for ± 100 kV charge voltage. (d) The AK gap insulator stack voltage and capacitor voltage reversal for ± 100 kV.....	106
Figure 5.4	(a) The foil load current and voltage ($\times 10$) performance for ± 70 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.....	108
Figure 5.5	(a) The foil load current and voltage ($\times 10$) performance for ± 80 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.....	109
Figure 5.6	(a) The foil load current and voltage ($\times 10$) performance for ± 90 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.....	111
Figure 5.7	(a) The foil load current and voltage ($\times 10$) performance for ± 100 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.....	112
Figure 5.8	The predicted instantaneous MRT growth rates and integrated e-foldings for (a) ± 70 kV, (b) ± 80 kV, (c) ± 90 kV, and (d) ± 100 kV charge voltage. The Rayleigh-Taylor and generalized 3-region scalings are shown to illustrate the damping effect from the magnetic field.....	113
Figure 6.1	A preshot image of a foil load offset from the midplane. The prescribed anode spacing is shown in red.....	117
Figure 6.2	(a) The initial foil position distribution for the large offset case. (b) The initial observable foil thickness for the large offset case as viewed by the laser backlighter.....	118
Figure 6.3	The image times plotted onto their corresponding drive current waveforms for the large offset case.....	120
Figure 6.4	The temporal evolution of the foil plasma for the large offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.....	128
Figure 6.5	The continued temporal evolution of the foil plasma for the large offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after than camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow	

	respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.	129
Figure 6.6	(a) Lineouts extracted parallel to the left side of the foil from shot 309. The ablation fingers create the oscillatory density pattern. (b) The Fourier transform of part (a) showing the strong 0.4 mm wavelength. (c) Lineouts extracted parallel to the right side of the foil from shot 306. The ablation structure appears superimposed onto an enveloping structure with approximately a 1 mm wavelength. (d) The Fourier transform of part (c).	130
Figure 6.7	A sample preshot image for the small offset case. The prescribed anode spacing is shown in red.	132
Figure 6.8	(a) The initial foil position distribution for the small offset case. (b) The initial observable foil thickness for the small offset case as viewed by the laser backlighter.	133
Figure 6.9	The image times plotted onto their corresponding drive current waveforms for the small offset case.	134
Figure 6.10	The temporal evolution of the foil plasma for the small offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.	135
Figure 6.11	The continued temporal evolution of the foil plasma for the small offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.	136
Figure 6.12	(a) An annotated image and (b) corresponding current drive plot for a large offset case shot. (c) An annotated image and (d) current for a small offset case shot at the same time and current. In (a) and (c), the positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%,	

	mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.....	138
Figure 6.13	(a) Example perpendicular lineouts from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side lineouts for a small offset shot.....	140
Figure 6.14	(a) Example parallel lineouts from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side parallel lineouts for a small offset shot. The distances were relative to the top of the ROI.	143
Figure 6.15	(a) Example parallel lineout Fourier transform spectra from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side parallel wavelength spectra for a small offset shot. The spectra were individually normalized.	144
Figure 6.16	Expansion measurements referenced to the initial observable foil position for (a) the left side of the large offset case, (b) the right side for the large offset case, (c) the left side for the small offset case, and (d) the right side for the small offset case. Empirical best fit curves are shown for the expansion phase with a 38 ns delay time.	146
Figure 6.17	(a) Expansion measurements of the bulk plasma between the mean laser transmission thresholds for the large offset case. (b) Location of the bulk plasma center for the large offset case. (c) Expansion measurements of the bulk plasma between the mean laser transmission thresholds for the small offset case. (d) Location of the bulk plasma center for the small offset case.	150
Figure 6.18	MRT growth rate figure of merit measurements for: (a) the left side of the large offset case, (b) the right side for the large offset case, (c) the left side for the small offset case, and (d) the right side for the small offset case. Empirical best fit curves are shown to measure the exponential growth rates for all cases.	152
Figure 6.19	The predicted foil kinematics according to the ESM, along with actual measured initial positions, for (a) the large offset case, and (b) the small offset case. The resulting magnetic fields from the drive current and load inductance for (c) the large offset case, and (d) the small offset case. The predicted instantaneous growth rates and integrated e-foldings according to the Rayleigh model and the generalized model for (e) the large offset case, and (f) the small offset case.	156

LIST OF TABLES

Table 1	Operating pressures for the LTD module and trigger spark gap switches for various charge voltages.....	68
Table 2	Parameters for ray transfer matrices to estimate refraction due to foil plasma.....	125
Table 3	Ratios between the left side initial velocity and the right side initial velocity from empirical fit parameters for the bulk plasma expansion measurements.....	148
Table 4	Ratios between the left side acceleration and the right side acceleration from empirical fit parameters for the bulk plasma expansion measurements.....	149
Table 5	Initial amplitudes, growth rates, and corresponding e-folding times for a given foil side and offset.	153
Table 6	A comparison between the acceleration coefficients and the acceleration required to generate the measured MRT growth from the Rayleigh scaling.....	159

LIST OF APPENDICES

Appendix A Experimental Data	169
A.1 Azimuthally Correlated Ablation – Small Interwire Gap	169
A.2 Azimuthally Correlated Ablation – Large Interwire Gap	172
A.3 MRT – Large Foil Offset.....	177
A.4 MRT – Small Foil Offset.....	182
Appendix B Matlab Postprocessing Code	192
B.1 Azimuthally Correlated Ablation.....	192
B.2 MRT Foils	219
B.3 Load Traces	352
B.4 Load Parameters (MRT).....	353
B.5 Load Parameters (Correlated Ablation)	354
B.6 Load Images	355
B.7 Find Wire Lineout.....	356
B.8 Generate Pairs	357
B.9 Integrate with Array Reduction	358
B.10 Integrate with Leading Zeros.....	358
B.11 FFT 1D	359
B.12 Find Peak Time	359
Appendix C Circuit Model.....	361
C.1 Switch.....	361
C.2 Brick.....	361
C.3 Quadrant	362
C.4 Module	363
C.5 Premagnetizer	363
C.6 Assembly	364
Appendix D Foil Cartridge Design.....	367
Appendix E Equipment Information	371
E.1 DAQ	371
E.2 Relay Control Unit	373
E.3 Voltage Monitor Calibrations	374
Appendix F MRT Experimental Procedures.....	375
F.1 Shot Checklist	375
F.2 Settings	378

CHAPTER 1

INTRODUCTION

Numerous fields of study require a detailed understanding of matter in the plasma state at densities that may approach, or even exceed, that of solid metals, such as investigations of stellar cores, thermonuclear fusion energy source development, and various defense applications. One particular application, thermonuclear fusion, occurs when nuclei of light elements are fused together to release large amounts of energy, the same process that heats stellar interiors. The nuclei of these elements are positively charged and are consequently repelled from each other by the Coulomb force, but if these nuclei can be brought sufficiently close together, the very short ranged strong nuclear force causes the nuclei to attract and fuse together, releasing significant energy in the process. The nuclei must have sufficient kinetic energy to overcome this Coulomb barrier, which for thermonuclear fusion is supplied by heating the nuclei to temperatures exceeding 50 million degrees Kelvin. At such high temperatures the fusion fuel must exist in the plasma state where the nuclei are ionized. Each fusion reaction releases an amount of energy, Q , at a reaction rate, $\langle\sigma v\rangle$, for a given thermal energy, kT . The energy released from the fusion reactions to sustain the high ion temperature requires:

$$n t > \frac{12kT}{\langle\sigma v\rangle Q}, \quad 1.1$$

where n is the ion number density and t is the confinement time. The high energy density inside a fusion plasma forces the fuel to explode and prevents it from being directly

confined by any container, so the confinement time represents how long the fusion plasma can remain intact before it blows itself apart and cools to end the fusion reaction chain. Equation 1.1 describes the minimum threshold for the product of the density and confinement time for a given fusion reaction at a given temperature. This is known as the Lawson criterion. Although many factors complicate the conditions for a fusion reaction to be self sustaining, the underlying result from the Lawson criterion remains that a fusion plasma must be sufficiently dense and confined for a long enough period of time in order to produce a net energy gain. One method for increasing the confinement time is to use strong magnetic fields to hold the plasma while it fuses and releases energy. This approach is known as magnetic confinement fusion (MCF). Another method is to simply rely on the fuel's own inertia to hold the hot plasma in place long enough for a fusion burst to occur before the fuel can blow apart. This approach requires much greater density than MCF due to the negligible confinement time, and is known as inertial confinement fusion (ICF).¹

The inertial confinement concept revolves around placing fusion fuel, usually a cryogenic deuterium and tritium mixture, inside a millimeter-scale spherical capsule. The capsule shell is typically made of hydrocarbons or beryllium. This outer shell is then rapidly heated using X-rays, lasers, or particle beams, which forces the surface to explode outward. The reaction force on the capsule interior causes the remaining shell, called the pusher, to implode and compress the fusion fuel. The conditions inside the imploding capsule can reach 10 keV ion temperatures and densities hundreds of times greater than solid hydrogen. A hotspot forms inside the fuel and ignites the fusion reaction that “burns” through the remaining fuel before the capsule explodes on a nanosecond

timescale. A limiting factor for ICF is the symmetry required during the implosion. If the capsule implodes asymmetrically, the fusion fuel does not reach sufficient temperature and density to ignite and the capsule fizzes. A hydrodynamic instability, known as the Rayleigh-Taylor (RT) instability, forms on the imploding capsule surface and attempts to break apart the pusher. If this instability grows violently enough, the implosion symmetry will be lost and the capsule will not ignite.² Additionally, one device used to generate the X-ray drive for imploding an ICF capsule, the wire array Z-pinch, suffers from a generalized version of the same instability.

A wire array Z-pinch, described in detail in Chapter 2, operates by passing millions of Amperes through a cylindrical array of thin metallic wires. The wires heat and ablate into plasma that implodes toward the central axis due to the self-generated magnetic pressure. The wire plasma pinches into a hot, dense column that emits X-rays useful for imploding an ICF capsule. The outer edge of the imploding wire plasma develops the magneto-Rayleigh-Taylor instability, which disrupts the cylindrical implosion symmetry and is a contributing factor, along with “sausage” and “kink” modes, that reduces the uniformity of the pinch column, resulting in a degraded X-ray burst. One source of this instability is the non uniform manner in which the wires ablate into plasma. The uniformity of the wire ablation process therefore plays a vital role in determining the X-ray performance from a Z-pinch; the uniformity of these ablation dynamics is the focus of Chapter 2.

The complex behavior of Z-pinch plasmas is often modeled as a single conducting fluid responsive to magnetic fields. This generalized branch of fluid mechanics is referred to as magnetohydrodynamics (MHD), and is a valuable framework for studying the

behavior of these plasmas. For a conducting fluid, the hydrodynamic Rayleigh-Taylor instability becomes the more general magneto-Rayleigh-Taylor (MRT) instability, which includes magnetic pressure effects in addition to kinetic pressure effects. This instability is classically described by placing a sheet of heavier fluid, such as water, above a lighter fluid, such as oil, in the presence of a gravitational field. The heavier fluid tends to fall below the lighter fluid, but cannot do so if the interface is perfectly flat, and the system will remain in an unstable equilibrium. If the interface is perturbed, the lighter fluid will rise as bubble-like structures and the heavier fluid will fall as spike-like columns. These bubbles and spikes initially grow exponentially with time and break apart the unstable interface. A generalized theoretical treatment of the MRT instability with three regions of arbitrary density and parallel magnetic field amplitude in a planar geometry is presented in Chapter 3, including the generalized dispersion relation, instability coupling between the two interfaces, and the anisotropy in instability evolution due to the magnetic field. A three region model is of particular interest compared to a two region model because it can be used to model an accelerating plasma liner surrounded by magnetic fields, such as the imploding liners used for the magnetized-target fusion concept.^{3,4}

The MAIZE (Michigan Accelerator for Inductive Z-pinch Experiments) facility at the Plasma, Pulsed Power, and Microwave laboratory at the University of Michigan was constructed and used to experimentally investigate the MRT instability. Aluminum foils 400 nm thick were used as loads. These foils were ablated into plasma which was unstable to the MRT instability and the growth rate was measured. The experimental setup is described in Chapter 4 and the results presented in Chapter 6.

The pulsed power device at the heart of the MAIZE facility was a linear transformer driver (LTD).⁵⁻⁸ The architecture for this LTD system represents the state of the art in present pulsed power system design. A circuit model of the MAIZE LTD is presented in Chapter 5. The LTD architecture consists of subunits, commonly referred to as bricks, arranged in parallel to form a larger unit cell referred to as an induction cavity or module. These modules are frequently pillbox-shaped toroids with a continuous azimuthal anode-cathode (AK) gap filled with an insulating dielectric. The output voltage from a module appears across this AK gap. The open central region of the toroids are typically sealed using vacuum flanges and evacuated to pressures below 10^{-4} Torr. The parallel brick configuration serves to add the current from each subunit to produce a large module current. The inductance of each brick is minimized to produce a fast-rising current pulse without the need for the pulse forming lines required for Marx-based systems of comparable specifications.⁹ The elimination of a pulse forming network and the low inductance configuration results in a pulsed power driver that consumes only a fraction of the volume an equivalent Marx-based system would require. The modules are then connected in series to add the voltages from each module, typically by inducing the voltage from each module onto a central stalk that is inside the evacuated central region and coaxial with the stacked modules in the same manner as for inductive voltage adder machines.¹⁰ This inductive voltage addition method reduces the electric field stresses across each module AK gap because the AK gap insulator only needs to hold off its own output voltage. The total machine output voltage only appears across the central stalk with respect to ground, as opposed to a Marx-based system where the final series capacitive element becomes electrically floated to nearly the total machine output

voltage. This allows for only a small volume of insulating oil to be used inside each LTD module and for a smaller AK gap insulator. The result of this unit cell design is a scalable architecture suitable for constructing next generation petawatt-scale pulsed power drivers.¹¹

One additional feature of the LTD architecture is the use of ferromagnetic cores, again similar to those used on inductive voltage adder systems. These cores allow the output electrodes to be grounded to the machine casing with the load in parallel with this ground path. Without the cores, the machine simply fires into the grounded casing, but the presence of the cores adds a large nonlinear impedance to this current path and nearly eliminates this parasitic current loss for fast pulses, forcing the main drive to pass through the load. The capacitor output buses remain firmly grounded whenever the machine is not firing. This results in a passively-grounded high voltage system for improved safety.

CHAPTER 2

AZIMUTHALLY CORRELATED ABLATION BETWEEN Z-PINCH WIRE CORES

Wire-array z-pinch experiments on Sandia National Laboratories' 'Z' accelerator, delivering >20 MA with a 100 ns risetime to hundreds of wires, routinely produce 2 MJ X-ray yields exceeding 200 TW in pulses¹²⁻¹⁴ suitable for K-shell X-ray production,^{12,15,16} inertial confinement fusion,^{2,17,18} and laboratory astrophysics.¹⁹ One limiting factor in the production of these powerful X-ray bursts is the growth of a disruptive magneto-Rayleigh-Taylor (MRT) instability in the imploding plasma seeded by non-uniform ablation dynamics from the wires prior to implosion.^{17,20,21} This critical coupling makes the understanding of the wire ablation dynamics crucial for improving Z-pinch X-ray performance.

Z-pinch wires do not uniformly ablate into plasma, but rather form an initial inhomogeneous, $10^{21}/\text{cm}^3$ number density resistive core of volumetrically boiling liquid and vapor states²²⁻²⁴ surrounded by $10^{18}/\text{cm}^3$ conductive local coronal plasma²⁵ that shunts the driver current away from the wire core. The hot corona slowly ablates the core, resulting in wire cores that persist until a cylindrical wire-array implodes.²⁶ This ablation is axially non-uniform where μm -scale gaps develop in the core and widen with time.²⁵

In single wire experiments, the axial-coronal structure becomes dominated by an $m=0$ instability.²⁷ Axial regions of coronal plasma necking increase the current density,

and consequently the electron temperature, near the dense core, resulting in regions with enhanced core ablation.^{28,29} Mass from these regions is axially transported into coronal bulges until the core runs out of material and a millimeter-scale gap structure forms.³⁰ When the ablating wire is part of an array, the global magnetic field removes coronal material in “finger-like” plasma streamers emanating from the bulging coronal regions towards the central axis.²⁶ The wire core radii reach an approximate equilibrium^{23,25} and the axial locations of plasma streams settle into a material dependent characteristic axial mode.^{26,31,32} Qualitative observations from several low-wire-number experiments suggest that the axial phases of these ablation stream locations are random, i.e. they are azimuthally uncorrelated.^{21,32}

High-wire-number arrays are most frequently used on large drivers, as increasing the number of uncorrelated perturbations from characteristic mode wire ablation decreases MRT growth and dramatically improves X-ray performance³³ until an optimal wire number is reached,^{34,35} beyond which X-ray performance degrades. At the same time, qualitative observations from experiments on the 1 MA MAGPIE accelerator⁹ indicated a possible shift in azimuthal correlation behavior where imploding magnetic bubbles, which have been shown to correlate with axial locations of wire-core gaps, appeared to azimuthally correlate between three neighboring wires in a 32 wire array (1.67 mm interwire gap), an effect not observed in experiments with 16 wires (3.32 mm interwire gap) or less on the same generator.³⁶ Similar behavior was observed on ‘Z’ where X-ray backlighter transmission through 5-7 wires implied azimuthal correlation between imploding magnetic bubbles over 1.67 mm, as well as the presence of a trailing mass structure with a 6-8-mm scale length in the azimuthal direction.^{37,38} It has been

suggested that the nature of this partially correlated system, with the resulting MRT instability coupled through the implosion into the stagnated pinch column, is one possible factor leading to the eventual falloff in X-ray performance for the highest wire numbers tested.³⁹ An attempt to influence the characteristic mode in low-wire-number arrays using chemically-etched density modulations was shown to generate coronal expansion flares at the seeding wavelength, which was 6-12 times longer than the characteristic mode, correlated with the initial modulation locations, as well as core gaps at the density discontinuities due to magnetic field enhancement. The characteristic mode structure, however, was superimposed onto the seeded structures and remained azimuthally uncorrelated.⁴⁰ Efforts to control the characteristic wavelength and azimuthal correlation were successful using low-wire-number coiled arrays where ablation streams formed at the coil wavelength and the azimuthal correlation was controlled by the initial coil phases.⁴¹ Despite the numerous advantages offered by coiled arrays, this configuration to date has not been fielded beyond ~1 MA accelerators.

Ablating wire cores have been computationally modeled as a mass injection boundary condition^{39,42} due to the complex nature of the ablation dynamics requiring μm -scale resolution over tens of millimeters in order to accurately model the core/corona structure for a 'Z' array containing hundreds of wires. The mass injection cells, however, must adequately mimic the true wire core plasma sources in order to eventually produce predictive results. Consequently, one key parameter for these mass injection cells is the azimuthal correlation fraction.³⁹

The purpose of this investigation was to provide insight into azimuthally-correlated ablation between wire cores through direct experimental measurement in both

a low-wire-number regime and an approximated high-wire-number regime. Data from two distinct experimental configurations on the 1 MA COBRA accelerator⁴³ were obtained to investigate the degree of correlation in the dense wire core gap structure between wires of differing interwire gaps in a cylindrical wire-array geometry.⁴⁴

2.1 Small Interwire Gap Experimental Configuration

The first configuration, shown in Figure 2.1, was designed to approximate the 210 μm interwire gap used for a 300 wire, 20 mm diameter array on ‘Z’. COBRA could not provide sufficient current to drive a 300 wire array, so a specialized wire configuration was constructed to approximate the conditions seen by two neighboring wires in a larger ‘Z’ array.

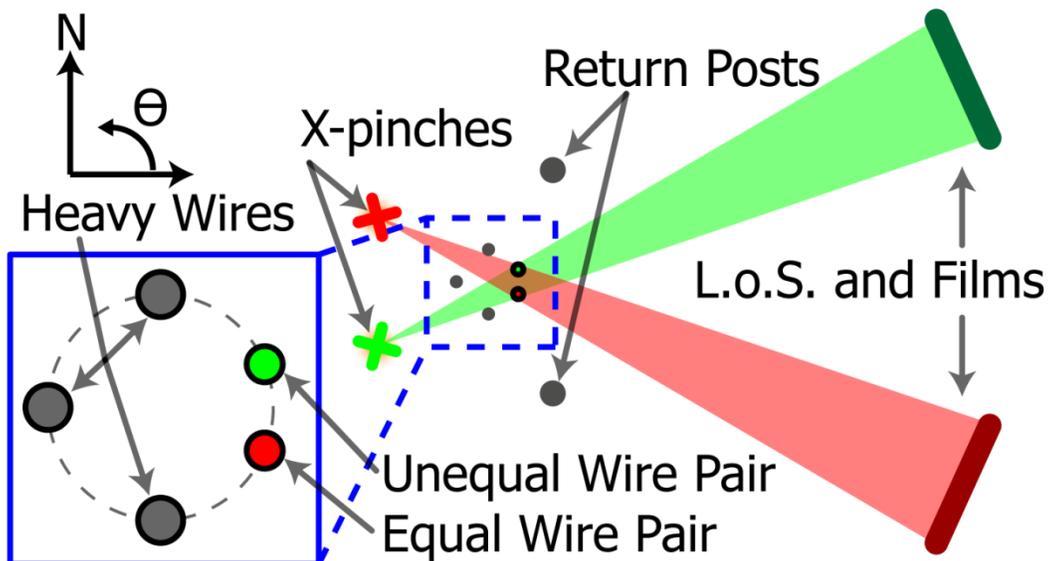


Figure 2.1 Wire locations and point-projection lines-of-sight for the closely-spaced wire configuration (not to scale). The south X-pinch (green) was timed to image the wire pairs earlier, while the north X-pinch (red) was timed to image the pairs later in time. The laser backlighter beams (not shown) passed through the array from left to right.⁴⁴

The 2 cm tall, 16 mm diameter tungsten array consisted of two closely-spaced wire pairs: an unequal-wire-diameter pair at $+22.5^\circ$ azimuth, and an equal-wire-diameter pair at -22.5° azimuth. Wire spacing within each pair was $240 \pm 30 \mu\text{m}$. The final shot in

this series also employed a slanted wire pair with 130 μm and 430 μm gaps at the bottom and top respectively. The resulting 0.43° slant was chosen to allow multiple interwire gaps on a single shot without drastically changing the wire dynamics. In order to produce a global magnetic field similar to that of a cylindrical array, 25 μm diameter wires were placed at 90° , 180° , and 270° . Two Mo X-pinch^{39,40} each consisting of 4 wires, were placed in the return current path 74.7 mm from the axis, the “north” X-pinch at 150° , and the “south” X-pinch at -150° . The X-pinch^{25,45} served as X-ray sources for point-projection radiography^{25,45} where each X-pinch imaged both wire pairs onto a single film.

Two lead-collimated film packs sensitive to 3-5 keV X-rays were used, each placed 50.3 cm from its corresponding X-pinch on the opposite side of the array. The film packs themselves consisted of a 6 μm mylar plus 12.5 μm Ti filter to isolate the 3-5 keV continuum X-rays from the X-pinch^{25,45}, followed by Kodak Industrex DR50 and Kodak BioMAX MS films. The higher contrast DR50 film images were used for the analysis. The resulting magnification was 7.1 for the closer pair and 6.8 for the farther pair on each image; the spatial resolution was $\sim 7 \mu\text{m}$ from the <1 -ns-duration X-ray bursts. The wire diameters in the unequal pair were 5.1 μm and 7.4 μm , whereas the diameters in the equal pair were both 7.4 μm . The results presented here concerned only the equal wire pair. A sub-millimeter scale epoxy drop was placed in the middle of the equal diameter pair to serve as an image registration fiducial on some shots. A three frame, 532 nm Nd:YAG laser backlighter with 200 ps pulse widths was also employed along the $180^\circ - 0^\circ$ axis with 0° , $+3^\circ$, and -1.5° exit angles to view the lower density coronal plasma from the wire pairs.

2.2 Large Interwire Gap Experimental Configuration

The second experimental configuration utilized the five frame COBRA-STAR point-projection radiography system⁴⁶ shown in Figure 2.2, to investigate azimuthally-correlated ablation between wires in low-wire-number, evenly-spaced arrays with large interwire gaps compared to those used on 300-wire ‘Z’ arrays.

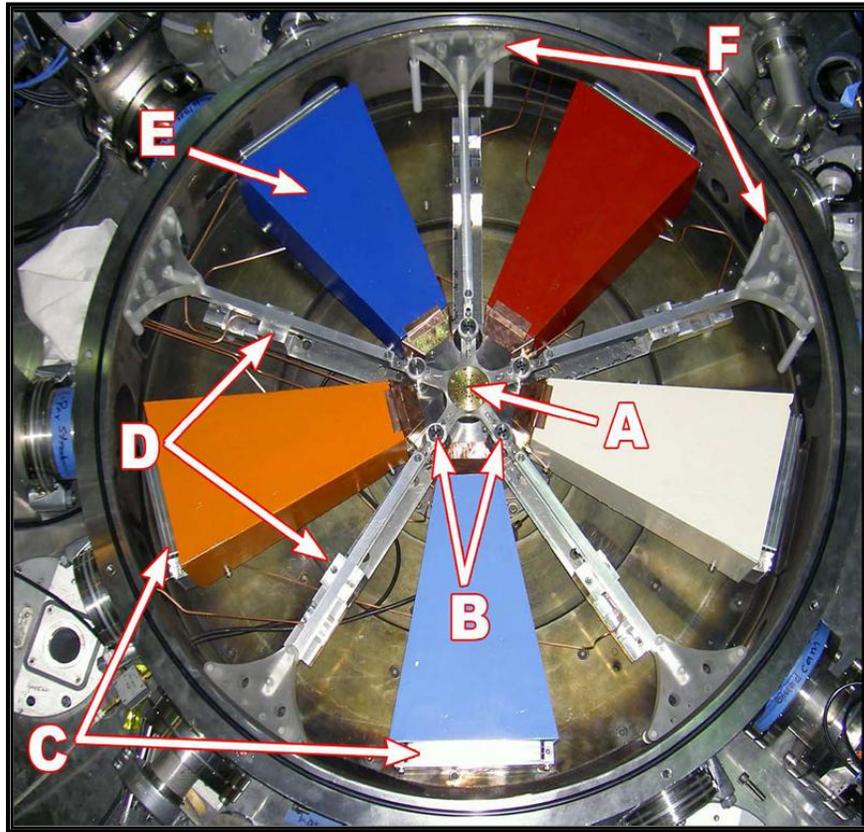


Figure 2.2 Experimental configuration for the COBRA-STAR imaging system and wire array load. The Z-pinch (A) was located in the center and was imaged by 5 X-pinch (B) in the current return path onto film packs (C). X-ray sensitive PCD detectors (D) provided timing information. Lead collimation (E) preserved film contrast and plastic supports (F) held up the system anode.⁴⁶

These 8 mm tall arrays consisted of 10 tungsten wires. The wires in a given array were all the same diameter, although different wire diameters were fielded on different shots.

Both 8 mm and 12 mm array diameters were used, with the corresponding straight line interwire gaps being 2.47 mm and 3.71 mm respectively. The COBRA-STAR system

was used to obtain multiple 3-5 keV X-ray backlit images of the entire array on each shot. Five Mo X-pinch wires of differing wire diameters were evenly placed around the array at a 67 mm radius. Each X-pinch produced an image onto a lead-collimated film pack located directly across the array at a 380 mm radius, resulting in an average magnification of 6.7 with $\sim 7 \mu\text{m}$ spatial resolution. The wires for the 8 mm and 12 mm diameter arrays were clocked 24° and 14° counter-clockwise respectively from the 0° ray to limit wire overlap on the images. A CH monofilament was placed at the center of the array to quench any stagnated Z-pinch column X-rays to preserve film contrast.

2.3 Image and Correlation Analysis

The scanned, high resolution X-ray films were analyzed for correlation using a suite of customized Matlab algorithms to extract lineouts along the Z-pinch wires, yielding intensity plots proportional to wire core plasma column density as a function of axial location. Wires were not perfectly straight when viewed on a scale similar to the core diameter, so a lineout generation algorithm was employed to extract lineouts of the wire cores unaffected by small wire displacement. The data processing and analysis algorithm is provided in Appendix B.

2.3.1 Wire Core Lineout Extraction Method

For a wire oriented in the z-direction with the y-direction primarily transverse to the wire in the plane of the image, the endpoints (x_1, y_1) and (x_2, y_2) of each wire section were recorded, and a y-direction search window, W , was specified such that the largest displacement of the wire from a straight line connecting the endpoints was encompassed within W . The y-direction pixel width of the wire, T , was also recorded. The lineouts were extracted by averaging the brightest $T/2$ pixels from each y-direction vector at each

z-direction location. This extraction geometry is illustrated in Figure 2.3. The resulting lineouts were therefore proportional to the average column density of the densest wire plasma as a function of axial distance.

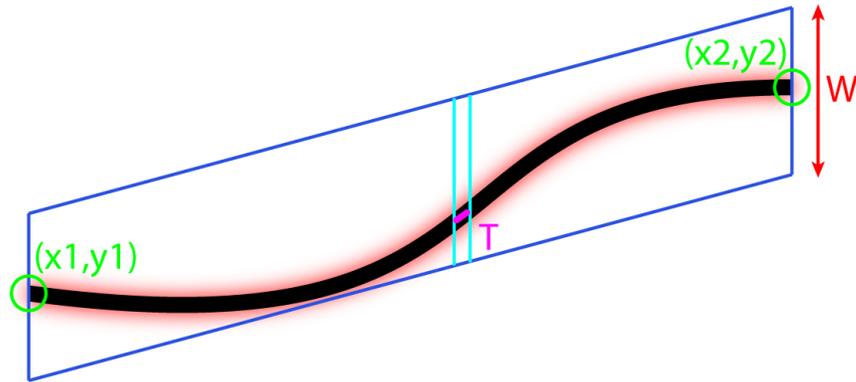


Figure 2.3 An illustration of the lineout extraction algorithm geometry.

The brightest pixels corresponded to the wire core plasma since the wire cores produced an approximately 50% smaller film response than the surrounding corona, except when the core was completely ablated away, in which case the lineout took on an upper limit for the coronal plasma value, or when a defect in the film with an outlying film response value existed within the search window. In order to remove outliers from film defects, a median filter was used where the filter window pixel width was chosen to be comparable to the resolution limit of the imaging system while still effectively removing the outliers. The region affected by epoxy from the fiducial on some small interwire gap shots was excluded from the analysis. The small interwire gap data also required as small a value for W as possible in order to exclude a dense local precursor plasma between the paired wires from the lineouts. The large interwire gap data set required an allowable magnification tolerance parameter to be chosen, which was set at 2% in order to limit magnification distortion at the lineout endpoints to the same order as the imaging

resolution. This parameter determined the maximum allowable difference in local magnification between wires in an array due to the finite array diameter for those wires to be considered suitable for comparison. Additionally, only the middle of the wire lengths was analyzed to avoid electrode effects. Some images from the large interwire gap data set also contained large scale ripple defects. Wires covered by such defects were either excluded from the analysis, or the lineout lengths across the entire image were reduced to preserve data integrity and to maintain uniform lineout length. The film regions extracted into lineouts were visually inspected to ensure that the automated algorithm successfully extracted the wire cores as intended.

2.3.2 Wire Pair Correlation Method

Paired lineouts were compared in both real space and in k-space by computing the correlation coefficient, R , between the lineout vectors \mathbf{L}_1 and \mathbf{L}_2 where $\langle \mathbf{L}_1 \rangle$ and $\langle \mathbf{L}_2 \rangle$ were the respective mean image intensity levels

$$R = \frac{(\mathbf{L}_1 - \langle \mathbf{L}_1 \rangle) \cdot (\mathbf{L}_2 - \langle \mathbf{L}_2 \rangle)}{\sqrt{\sum (\mathbf{L}_1 - \langle \mathbf{L}_1 \rangle)^2 \sum (\mathbf{L}_2 - \langle \mathbf{L}_2 \rangle)^2}} \quad 2.1$$

The correlation coefficient ranged between ± 1 where 1 indicated perfect linear correlation, -1 indicated perfect linear anti-correlation, and 0 indicated no correlation between the lineouts. Statistical significance for lineouts with n points was determined using a t-test with $n-2$ degrees of freedom and a 95% confidence threshold with zero correlation as the null hypothesis. The lineouts typically contained between 400 and 2000 data points so an asymptotic normal distribution was used to calculate the 95% confidence boundaries.

2.4 Experimental Results

The experimental results described in Section 2.4.1 for the small interwire gap, and in Section 2.4.2 for the large interwire gap, provided quantified proof of correlation in ablation structure between wire cores that was interwire gap dependent. The wires were observed to ablate with a similar spatial mode structure for both cases. This analysis, summarized in Section 2.4.3, successfully identified a wire spacing transition region below which the ablative spatial mode structure between wires locked in phase. Wire spacings above the transition region did not result in the ablative spatial modes locking in phase.⁴⁷

2.4.1 Small Interwire Gap

Figure 2.4 shows a representative radiograph of an equal-wire-diameter pair from the closely-spaced data set at 93 ns and 1.1 MA drive current. The magnified inset contained 1.9 mm of the wires with enhanced contrast and false color added to bring out the detailed plasma structure.

Qualitative observation of the radiograph indicated a highly-correlated ablation pattern between the wires that is further supported in the overlaid lineouts below the insert. Additionally, a prominent local coronal plasma precursor column was observed between the wires modulated by what appeared to be an $m = 0$ instability with a dominant 600 μm wavelength, bulged regions 150-200 μm across, and necked regions approximately 90 μm across. Coronal plasma flares were also observed and appeared symmetric on either side of the local precursor, similar to those typically seen around ablating wire cores, although these flares were observed even where the wire cores had completely ablated away and appeared to be local precursor features. The flares occurred

at the 220 μm wavelength characteristic mode for tungsten, as well as the 600 μm wavelength mode seen modulating the local precursor, and an additional 300 μm

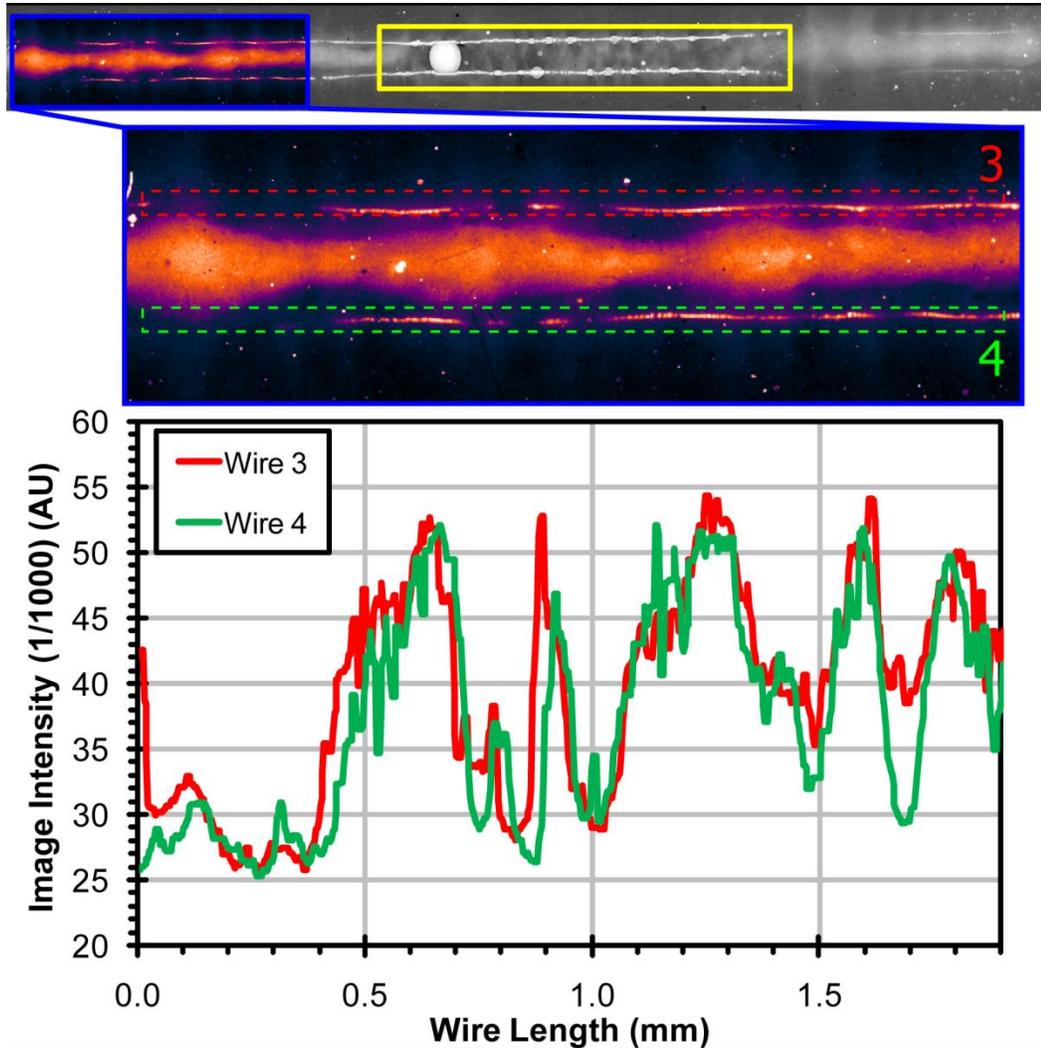


Figure 2.4 An enhanced contrast, false-color radiograph from shot 762 showing 1.9 mm of two 7.4 μm diameter tungsten wires spaced 240 μm apart at 93 ns and 1.13 MA on the cathode side of the fiducial region (yellow rectangle). The corresponding correlated lineouts are shown below.⁴⁴

wavelength mode. Every bulged region on the local precursor had a coinciding flare, although not every flare coincided with a bulged region within the 1.9 mm section in Figure 2.4. The yellow rectangle indicates the fiducial region where the wires were coated in epoxy. While excluded from the correlation analysis, this region showed

drastically different behavior from the bare wire regions. The bare wire regions did not appear to be affected by the fiducial region except that the immediately adjacent bare wires on the anode (right) side of the fiducial region appeared preferentially ablated. The wire cores within the fiducial region appeared intact and the local precursor was not nearly as prominent compared to the bare regions. Small coronal flares only originated from wire core regions that were less covered by the epoxy beads. This behavior suggested that the epoxy formed a hydrocarbon plasma that shunted the current away from the metallic cores and reduced the initial energy deposited into this region relative to the bare wires, and that the thicker beads of epoxy inhibited further energy transfer from the surrounding corona into the cores, causing a further reduction in ablation rate. The location of this local precursor was expected to be displaced towards the array axis rather than directly between the wires due to the global $\mathbf{J} \times \mathbf{B}$ force. The location could not be measured from the X-ray radiographs due to the lack of precursor formation between the equal pair wires in the earlier time images, and to the very small rotation between the X-ray backlighter and the wires in the later time images. A laser backlighter image 1 ns before the X-ray radiograph showed the plasma column 1.1 ± 0.3 mm from the plane connecting the equal pair wires, as well as a highly-correlated flare structure between the wire coronas. This laser image is shown in Figure 2.5. The presence of the $m=0$ instability on the local precursor strongly suggested that it was a current-carrying structure, although the current path would necessarily become complicated within the fiducial region where the column was less apparent, perhaps flowing through less dense hydrocarbon material approaching the sensitivity of the diagnostic. This correlated system and local precursor was similar to that observed in laser backlighter images by

Beg, *et al.*⁴⁸ during a series of two-wire Z-pinch experiments, but with the fundamental difference that, in this case, a global magnetic field was acting on the wire pair that was not present in those previous strictly two-wire shots. The most striking difference in

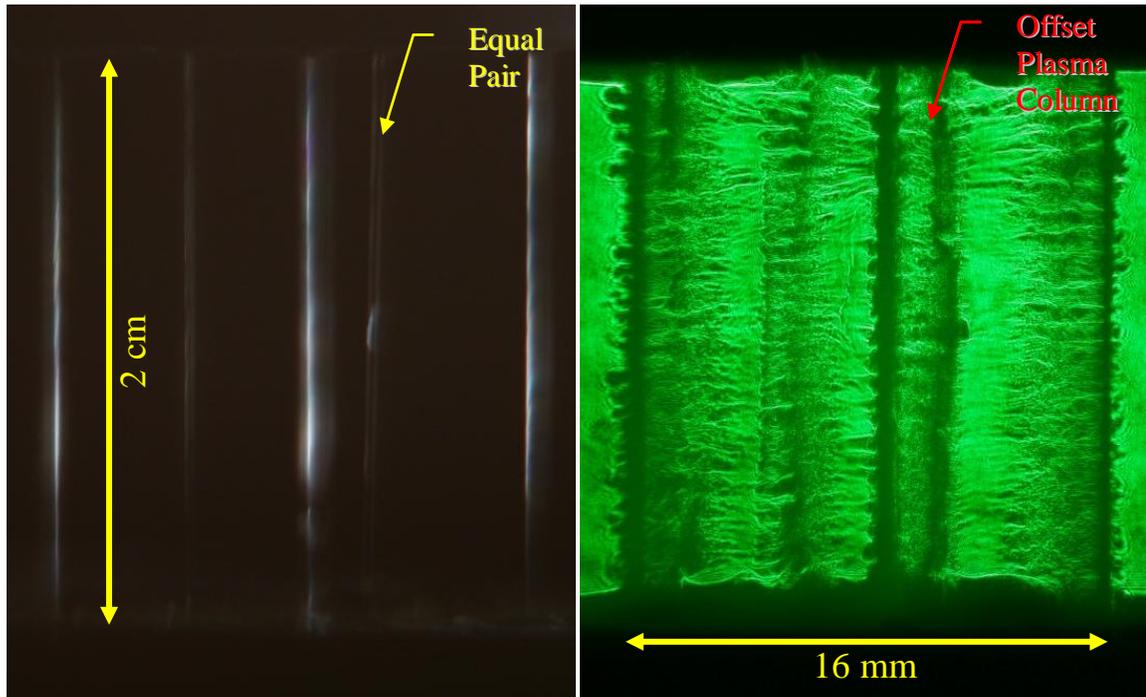


Figure 2.5 (a) A front-lit preshot image from laser path 3 from shot 762. (b) The data image at 92 ns from shot 762 showing the coronal plasma around the wire pairs, as well as the local precursor column displaced toward the axis from the wire plane.

behavior between these experiments was in the local precursor where the global magnetic field radially displaced the local precursor from directly between the wires towards the array axis, and the dominant $m = 1$ instability, seen in the two-wire case, was absent.

2.4.2 Large Interwire Gap

Figure 2.6 (a) shows one frame of ten evenly-spaced, $7.4 \mu\text{m}$ diameter wires taken at 100 ns and 1.0 MA. The analyzed portion of the image is again shown in false color and enhanced contrast. One of the wires was obscured by the X-ray-quenching monofilament. Lineouts from three wires at $6.37 \pm 2\%$ local magnification are overlaid in

Figure 2.6 (b). While these lineouts qualitatively showed a similar core ablation mode structure, the phase offset of this structure resulted in poor azimuthal correlation between the ablating cores.

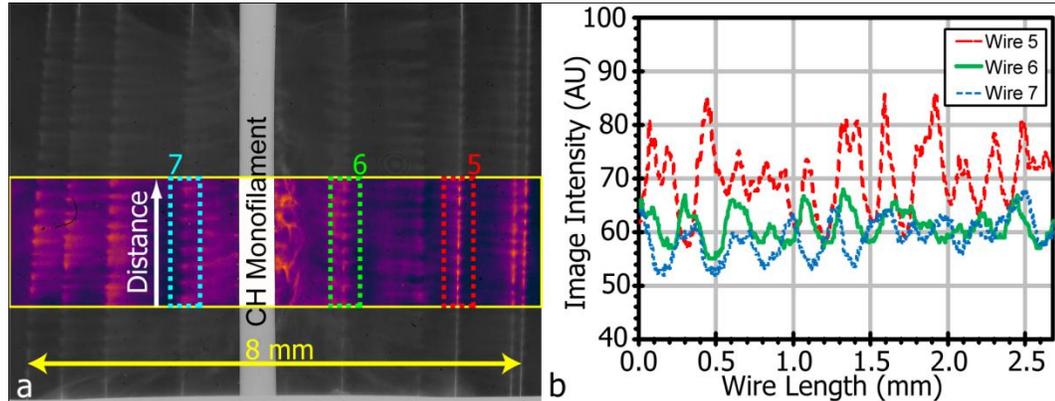


Figure 2.6 (a) An enhanced contrast radiograph from shot 800, taken using the COBRA-STAR imaging system of ten tungsten wires with $7.4\ \mu\text{m}$ diameters at 100 ns and 1.0 MA. The monofilament obscured one of the wires. Lineouts were extracted from the dotted regions within the false color portion of the image. (b) The corresponding lineouts from wires in the radiograph at $6.37\pm 2\%$ local magnification.⁴⁴

2.4.3 Experimental Results Summary

Figure 2.7 summarizes the quantified azimuthal correlation results from multiple frames from five closely-spaced shots, one closely-spaced shot with the paired wires slanted, and nine widely-spaced shots with various wire diameters and interwire gaps. The correlation coefficient, R , is plotted against the straight line distance between the compared wires in both real space and in k -space. In real space there was a striking difference in correlation, in agreement with qualitative observation. All measurements beyond 2.47 mm spacing showed randomly distributed correlation about zero, whereas the $240\ \mu\text{m}$ cases all displayed strong correlation with seven of the eleven equal-diameter pair measurements displaying $R > 0.75$, and all measurements displaying $R > 0.5$. In k -space the correlation was much stronger for all wire spacings, confirming that the ablation mode structure was similar between wires, but not necessarily correlated in

phase.⁴⁷ This result quantitatively proved the strong coupling in wire core ablation between closely-spaced wires in a global magnetic field and was consistent with qualitative observations.^{36,48} There was no observable change in correlation behavior from shot-to-shot or from wire diameter over the 7.4 μm to 12.7 μm range tested, nor was there any observable time dependence, although the limited intrashot time resolution was most likely insufficient on the closely-spaced configuration to observe any temporal correlation evolution.

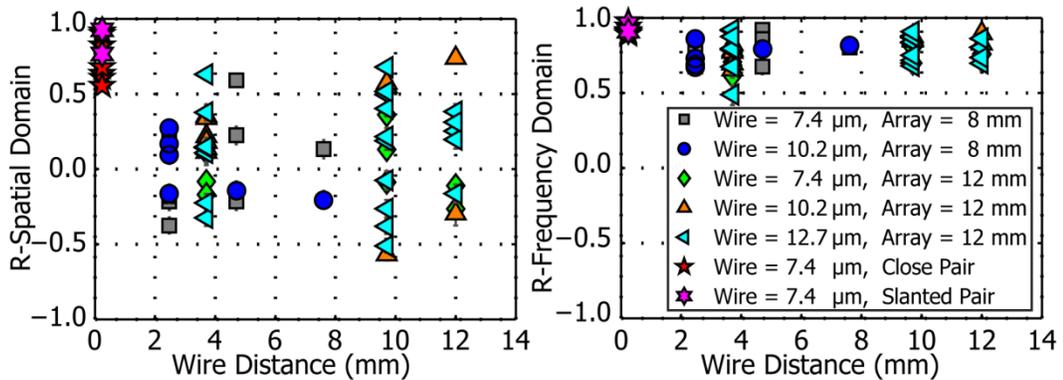


Figure 2.7 Correlation coefficient versus wire separation with 95% confidence intervals for various wire diameters and array diameters in the spatial domain (left) and in the spatial frequency domain (right) with the DC component excluded.⁴⁴

2.5 Physical Correlation Mechanisms

The experimental results were clear evidence that azimuthally-correlated ablation occurred between Z-pinch wire cores when the wires were placed more closely together than a critical separation threshold, determined to be between 0.24 mm and 2.47 mm. Two physical effects may have been responsible for this azimuthally-correlated ablation, both resulting in increased Joule heating in neighboring wire cores.

2.5.1 Azimuthal Current Shunting

One mechanism was current shunting through azimuthally-oriented current paths created by connected neighboring wire coronas. When a break in one wire generated an imploding magnetic bubble, these current paths allowed the current to shunt away from the more inductive bubble front into an inductively favorable path through a neighboring wire corona.³⁹ The current then shunted back to its original wire to maintain inductively-favorable parallel current paths. The additional current in the neighboring wire at the same axial location as the imploding bubble increased the Joule heating in that region, enhancing the ablation rate and leading to azimuthally-correlated wire core breaks. This mechanism is illustrated in Figure 2.8.

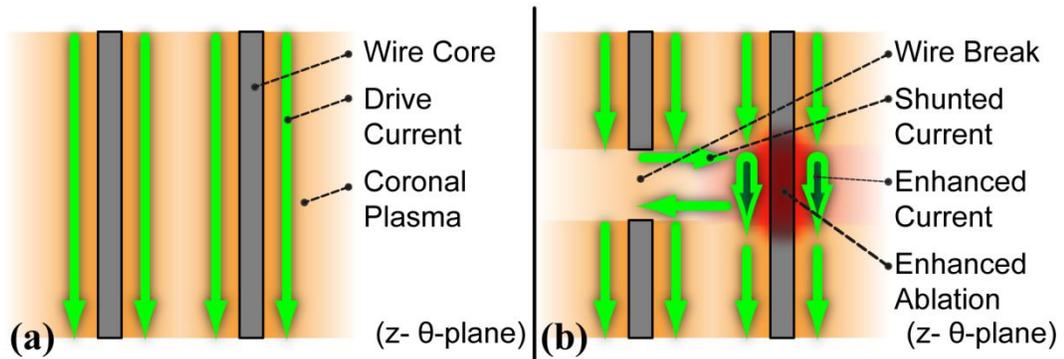


Figure 2.8 A current shunting mechanism for correlated ablation of two neighboring wires (a) prior to gap formation, and (b) after a break formed in left wire.⁴⁴

In Figure 2.8 (a) two neighboring ablating wire cores, which were part of a larger cylindrical wire array (other wires omitted) prior to gap formation, were viewed from the side ($z-\theta$ -plane) surrounded by conducting coronal plasma that carried the drive current. The same configuration is shown in Figure 2.8 (b) later in time after a break had formed in the left wire. The current shunted through an azimuthally-oriented current path around the imploding magnetic bubble through the connecting coronal plasma. The additional

current around the wire core on the right resulted in enhanced Joule heating and ablation correlated with the original break in the left wire.

2.5.2 Magnetic Compression

A second mechanism was the distortion of the global magnetic field near a wire core break. When the ablating wire core at a fixed axial location ran out of material, the coronal plasma at that location was swept towards the axis, creating an imploding magnetic bubble. The distorted magnetic field due to this bubble compressed the coronal plasma around a neighboring wire core at that same axial location, resulting in an increased ablation rate from Joule heating [Figure 2.9 (a, b)] and leading to azimuthally-correlated wire core breaks.

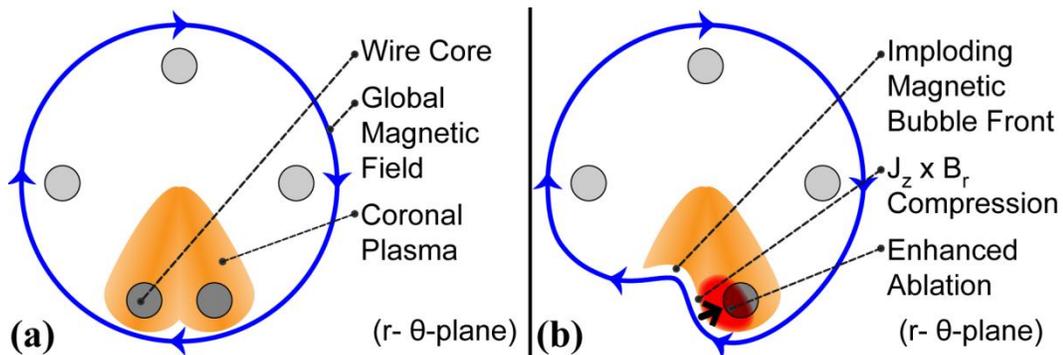


Figure 2.9 A magnetic compression mechanism for correlated ablation of two neighboring wires in the initial stage (a), and after break had formed in left wire (b). For clarity, only the plasma streamers from the lowest pair of wires were shown.⁴⁴

In Figure 2.9 (a), a wire array was viewed from above through an axial slice in z ($r-\theta$ -plane) with the drive current flowing into the page. The dominant global magnetic field topology from the entire array was shown, as well as ablation streams of coronal plasma from the paired bottom wires. Ablation streams from the rest of the array were suppressed for clarity. The same configuration was shown in Figure 2.9 (b) later in time after a break had formed in the left wire in the plane of the figure. A magnetic bubble

front was shown moving toward the axis, distorting the global magnetic field and creating a $J_z \times B_r$ force component that compressed the coronal plasma near the right wire. This compression resulted in enhanced Joule heating and ablation on the right wire correlated with the original break in the left wire.

CHAPTER 3

A GENERAL THEORETICAL TREATMENT OF THE MAGNETO-RAYLEIGH-TAYLOR INSTABILITY IN A PLANAR GEOMETRY

A theoretical framework describing the magneto-Rayleigh-Taylor (MRT) instability was a vital asset for planning, conducting, and interpreting experiments for measuring the instability growth rate. The general approach for such an analysis required solving the magnetohydrodynamic (MHD) equations by linearization and perturbation methods to obtain a dispersion relation for the instability. Additionally, the acceleration driving the instability needed to be understood. This analysis required solving the equation of motion for the interface in question. An idealized planar geometry with a thin foil positioned between return current plates is presented in Section 3.1. The acceleration experienced by a thin foil in this geometry is derived in Section 3.2. A derivation of the generalized MRT dispersion relation for this 3 region geometry is shown in Section 3.3. The concept of instability feed-through, defined as instability amplitude on an unstable interface imprinting onto a stable interface, is addressed in Section 3.4. The generalized dispersion relation was then used to recover the classical results of Rayleigh, Taylor,⁴⁹ Harris,⁵⁰ and Chandrasekhar⁵¹ in Section 3.5, as well as the case describing the geometry presented in Section 3.1. Much of the theoretical treatment of this chapter was based on a series of lectures given by Professor Y.Y. Lau.

3.1 Idealized Planar Foil Load Geometry

A planar geometry with double return plates was used as the model for the MRT foil load. The return plates were located at $x = \pm D$ from the midplane and the foil was placed between the return current plates at a displacement $x = x_0$. The foil had a mass density ρ , height h , thickness W , and length L . The drive current, I , passed through the foil and was shared between the return current plates as I_1 and I_3 . This geometry is shown in Figure 3.1. The subscript 3 was chosen instead of 2 since, later, in Figure 3.2, region 2 was reserved for the foil interior.

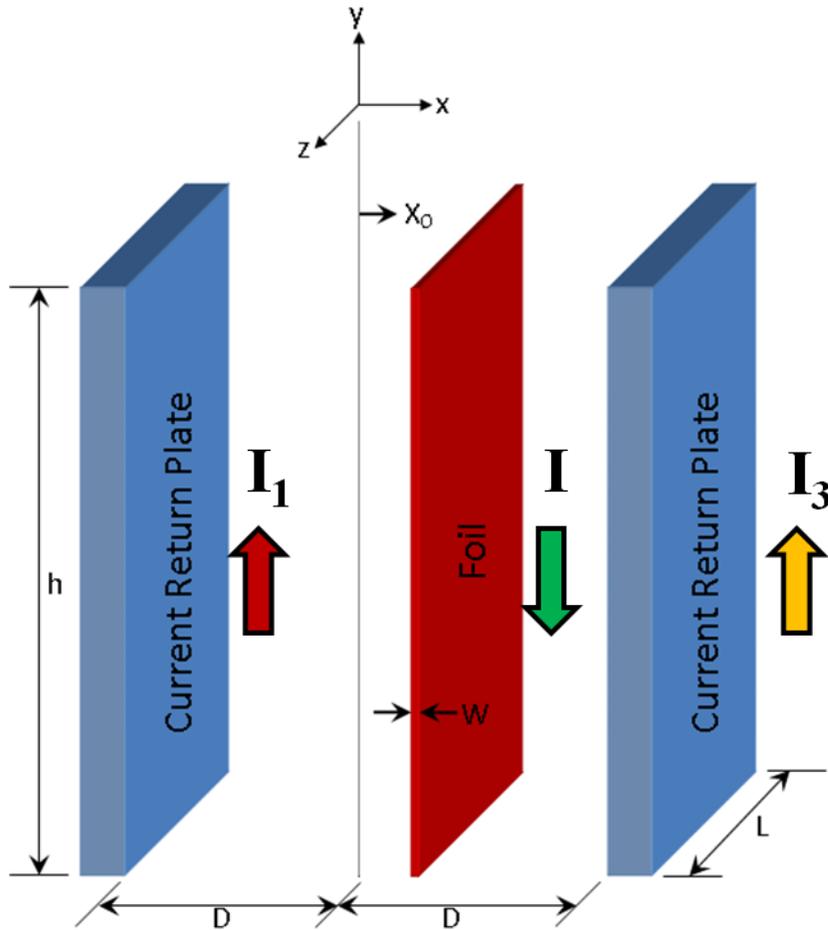


Figure 3.1 The geometry for the planar foil load region.

3.2 Derivation of the Acceleration for an Expansionless Slab

The acceleration experienced by the foil was determined from the equation of motion for the foil. In order to solve for this acceleration due to the $\mathbf{J} \times \mathbf{B}$ force, the current densities and the corresponding magnetic fields on each side of the foil needed to be found. Conservation of electrical charge required that the currents, or equivalently the surface current densities, flowing through the return current plates must be equal to the total drive current through the foil

$$I = I_1 + I_3, \quad 3.1$$

$$K = K_1 + K_3. \quad 3.2$$

Since the impedance division between the return paths was dominated by the changing magnetic flux term, particularly after the foil quickly formed a plasma of negligible resistivity compared to the solid foil, the ratio of current flowing through each return plate was approximated as being inversely proportional to the inductance of each return path

$$\frac{I_1}{I_3} = \frac{K_1}{K_3} = \frac{\Lambda_3}{\Lambda_1}, \quad 3.3$$

where Λ was the path inductance of a given return path, and K was the current density through the given return path. The location x was taken to be the instantaneous position of the foil and H to be the auxiliary magnetic field. The inductances for each path were therefore

$$\Lambda_1 = \frac{\phi_1}{I_1} = \frac{\mu_0 H_1 h}{I_1} \left(D + x - \frac{w}{2} \right) = \frac{\mu_0 h}{L} \left(D + x - \frac{w}{2} \right), \quad 3.4$$

$$\Lambda_3 = \frac{\phi_3}{I_3} = \frac{\mu_0 H_3 h}{I_3} \left(D - x - \frac{w}{2} \right) = \frac{\mu_0 h}{L} \left(D - x - \frac{w}{2} \right). \quad 3.5$$

The second simplification utilized the planar geometry of the system where $|\mathbf{H}| = |\mathbf{K}| = I/L$. The fast-rising current was also assumed to flow entirely on the lower inductance foil surfaces. Inserting Equations 3.4 and 3.5 into Equation 3.3 yielded

$$\frac{K_1}{K_3} = \frac{\left(D - x - \frac{w}{2}\right)}{\left(D + x - \frac{w}{2}\right)}, \quad 3.6$$

which was combined with Equation 3.2 to find the current densities on either side of the foil as a function of x and the total current density K

$$K_1 = K \left(\frac{D - x - \frac{w}{2}}{2D - w} \right), \quad 3.7$$

$$K_3 = K \left(\frac{D + x - \frac{w}{2}}{2D - w} \right). \quad 3.8$$

The equation of motion for the foil could then be used to determine the acceleration from the drive current pulse. The force law for the foil material was

$$\mathbf{a} = \frac{\mathbf{J} \times \mathbf{B}}{\rho}, \quad 3.9$$

where \mathbf{a} was the acceleration and \mathbf{B} was the average magnetic field. The magnetic fields on either side of the foil were given by

$$\mathbf{B}_1 = \frac{\mu_0 I}{L} \left(\frac{D - x - \frac{w}{2}}{2D - w} \right) (-\hat{\mathbf{z}}), \quad 3.10$$

$$\mathbf{B}_3 = \frac{\mu_0 I}{L} \left(\frac{D + x - \frac{w}{2}}{2D - w} \right) \hat{\mathbf{z}}. \quad 3.11$$

The total magnetic field in the interaction region between the return current plates was the superposition of the magnetic fields on each side of the foil. From Ampere's Law,

these fields were zero outside their respective regions, including inside the foil interior. Since $w / D \ll 1$, the total field at the negligibly wide foil was approximated as the averaged vector sum of Equations 3.10 and 3.11

$$\mathbf{B} = \frac{\mu_0 I}{2L} \frac{x}{D} \hat{\mathbf{z}}. \quad 3.12$$

Substituting Equation 3.12 into Equation 3.9 and using the relation $\mathbf{J} = \mathbf{K}/w$ allowed the foil acceleration to be determined

$$\mathbf{a} = -\frac{\mu_0 I^2(t)}{2\rho w L^2 D} x \hat{\mathbf{x}}, \quad 3.13$$

$$\frac{d^2 x}{dt^2} = -\frac{\mu_0 I^2(t)}{2\rho w L^2 D} x(t). \quad 3.14$$

Equation 3.14 clearly described a harmonic oscillator for the foil position as a function of time. However, the “spring constant” of this harmonic oscillator was time-dependent due to the drive current factor. The foil position needed to be calculated by numerical integration methods. The expansionless slab model (ESM) therefore consisted of the numerical solution to Equation 3.14 for $x(t)$, from which the total magnetic field was obtained using Equation 3.12. The foil velocity and position followed as integrals of the acceleration from Equation 3.13 for each time step. The magnetic fields on either side of the foil were also known from Equations 3.10 and 3.11 as a function of time. The acceleration scaled as the square of the drive current, which indicated that larger drive currents were particularly important for increasing the acceleration. The acceleration was inversely proportional to the foil density, width, and electrode spacing. These quantities either increased the foil mass or the load inductance, so increases in these quantities reduced the acceleration as expected. The acceleration also scaled as the inverse square of the foil length, L . This stronger dependence on the length, as compared to the foil

thickness or density dependence, was a result of several competing effects. Increasing L increased the foil mass, but this effect was canceled by the reduction in load inductance. The additional scaling resulted from the imposed drive current being spread out over the larger foil surface, which reduced the current density and created the inverse square dependence on L . The acceleration was not explicitly a function of the foil height, h ; however, the drive current from any physical generator that ultimately created the acceleration would have depended on the total load inductance, which did increase with the foil height. This model is further explored in Chapter 5.

3.3 Derivation of the Generalized Dispersion Relation

The idealized geometry shown in Figure 3.1 was modeled using three planar regions of infinite extent in the y - and z -directions separated by interfaces $x = \alpha$ and $x = \beta$. The distance between the interfaces was $\Delta = \alpha - \beta$. This geometry is shown below in Figure 3.2. Each region was characterized by an initial density, ρ_0 ; magnetic field, $\mathbf{B} = B_0 \hat{\mathbf{z}}$; pressure, $P_0(x)$; current density, $\mathbf{J}_0 = 0$; and velocity, $\mathbf{v}_0 = 0$. The acceleration, $\mathbf{g} = g \hat{\mathbf{x}}$, was the acceleration experienced in the *rest* reference frame of the interfaces. The ideal MHD equations for MRT in a slab of finite width for each region were

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mathbf{J} \times \mathbf{B} - \rho g \hat{\mathbf{x}}, \quad 3.15$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad 3.16$$

$$\nabla \cdot \mathbf{v} = 0, \quad 3.17$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad 3.18$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}, \quad 3.19$$

$$\mathbf{E} + \mathbf{v} \times \mathbf{B} = 0, \quad 3.20$$

where Equation 3.15 was the equation of motion, Equation 3.16 was the equation of continuity, Equation 3.17 was the incompressible fluid condition, Equation 3.18 was Faraday's Law, Equation 3.19 was Ampere's Law neglecting displacement current, and Equation 3.20 was Ohm's Law for a perfectly conducting fluid (ideal MHD with the magnetic field lines frozen into the plasma). The magnetic fields, oriented in the z -direction, were

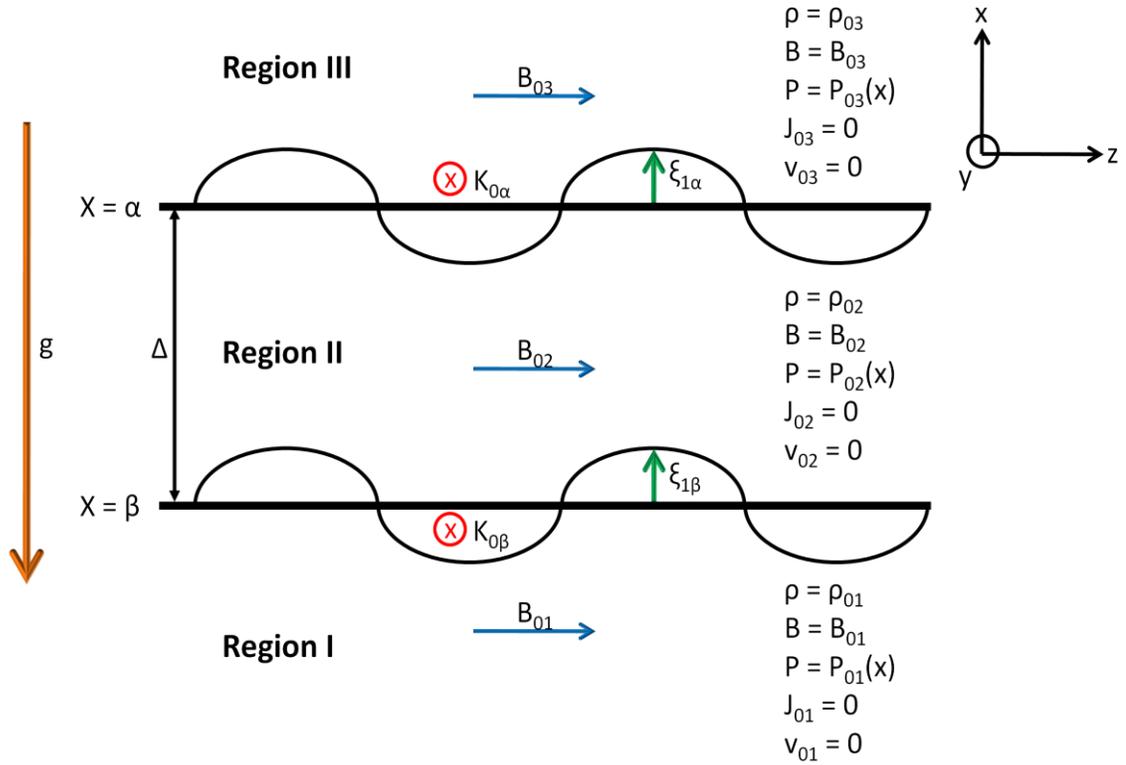


Figure 3.2 The geometry for the generalized three region MRT instability model.

allowed to be discontinuous across the interfaces. This discontinuity required the presence of a surface current K_0 at each interface according to

$$\mathbf{K}_{0\alpha} = \frac{(B_{02} - B_{03})}{\mu_0} \hat{\mathbf{y}}, \quad 3.21$$

$$\mathbf{K}_{0\beta} = \frac{(B_{01}-B_{02})}{\mu_0} \hat{\mathbf{y}}, \quad 3.22$$

where the second numerical subscript denoted the region for each field shown in Figure 3.2, and α or β denoted the interface containing the surface current.

3.3.1 Fluid Layer Equilibrium

The equilibrium conditions for the given geometry required that the total pressure, defined as the sum of the kinetic pressure and the magnetic pressure, was equal across the interfaces

$$\left[P_{03}(\alpha^+) + \frac{B_{03}^2}{2\mu_0} \right] = \left[P_{02}(\alpha^-) + \frac{B_{02}^2}{2\mu_0} \right], \quad 3.23$$

$$\left[P_{02}(\beta^+) + \frac{B_{02}^2}{2\mu_0} \right] = \left[P_{01}(\beta^-) + \frac{B_{01}^2}{2\mu_0} \right], \quad 3.24$$

where the pressures are to be evaluated at x-positions infinitesimally above (+) or below (-) the interface at $x = \alpha$ or $x = \beta$. Additionally, the weight of region 2 and the downward pressure from region 3 was supported by the upward pressure from region 1, allowing the region 2 fluid to float in place between the region 1 and 3 fluids

$$\left[P_{01}(\beta^-) + \frac{B_{01}^2}{2\mu_0} \right] = \left[P_{03}(\alpha^+) + \frac{B_{03}^2}{2\mu_0} \right] + \rho_{02}g\Delta. \quad 3.25$$

Equations 3.23, 3.24, and 3.25 explicitly showed how the source of the total pressure could have been kinetic, magnetic, or both as long as these equilibrium conditions remained satisfied.

3.3.2 Perturbation from Equilibrium and Linearization

A small disturbance in displacement, $\xi_1 = \xi_1(x, y, z, t)$, was next introduced on the equilibrium. The magnitude of the displacement perturbation was $\xi_{1\alpha}$ at the interface

$x = \alpha$, and $\xi_{1\beta}$ at the interface $x = \beta$ [Figure 3.2]. The quantities in Equations 3.15 - 3.20 therefore contained an equilibrium component, denoted by a subscript 0, and a perturbation component, denoted by a subscript 1. Terms containing only equilibrium quantities simply described the equilibrium conditions and were therefore ignored in perturbation analysis. Terms that contained products of perturbation quantities were second order and were also ignored. The remaining terms were those linear in perturbation quantities. The linearized MHD equations were therefore

$$\rho_0 \frac{\partial \mathbf{v}_1}{\partial t} = -\nabla p_1 + \mathbf{J}_1 \times \mathbf{B}_0 - \rho_1 g \hat{\mathbf{x}}, \quad 3.26$$

$$\frac{\partial \rho_1}{\partial t} + \rho_0 (\nabla \cdot \mathbf{v}_1) + \mathbf{v}_1 \cdot \nabla \rho_0 = 0, \quad 3.27$$

$$\nabla \cdot \mathbf{v}_1 = 0, \quad 3.28$$

$$\nabla \times \mathbf{E}_1 = -\frac{\partial \mathbf{B}_1}{\partial t}, \quad 3.29$$

$$\nabla \times \mathbf{B}_1 = \mu_0 \mathbf{J}_1, \quad 3.30$$

$$\mathbf{E}_1 + \mathbf{v}_1 \times \mathbf{B}_0 = 0. \quad 3.31$$

The functional form of the perturbation needed to satisfy Equations 3.26 - 3.31 and to decay with distance as $x \rightarrow \pm\infty$. From Equation 3.61 below, it will be shown that the perturbations were of the form

$$n_1 = [Ae^{k_x x} + Be^{-k_x x}] e^{-i(k_y y + k_z z + \omega t)}, \quad 3.32$$

where n_1 represented any of the perturbation quantities in each region: I, II, and III of Figure 3.2. This perturbation physically represented a periodic structure in the y- and z-directions that would exponentially decay in the x-direction with distance as $x \rightarrow \pm\infty$. Additionally, the perturbed velocity was expressed in terms of the perturbed displacement as

$$\mathbf{v}_1 = \frac{\partial \boldsymbol{\xi}_1}{\partial t}. \quad 3.33$$

Inserting Equations 3.32 and 3.33 into Equations 3.26 - 3.31 yielded

$$-\omega^2 \rho_0 \boldsymbol{\xi}_1 = -\nabla p_1 + \mathbf{J}_1 \times \mathbf{B}_0 - \rho_1 g \hat{\mathbf{x}}, \quad 3.34$$

$$\rho_1 + \rho_0 (\nabla \cdot \boldsymbol{\xi}_1) + \boldsymbol{\xi}_1 \cdot \nabla \rho_0 = 0, \quad 3.35$$

$$\nabla \cdot \boldsymbol{\xi}_1 = 0, \quad 3.36$$

$$\nabla \times \mathbf{E}_1 = i\omega \mathbf{B}_1, \quad 3.37$$

$$\nabla \times \mathbf{B}_1 = \mu_0 \mathbf{J}_1, \quad 3.38$$

$$\mathbf{E}_1 - i\omega \boldsymbol{\xi}_1 \times \mathbf{B}_0 = 0. \quad 3.39$$

From Equation 3.32 and Equation 3.61 below, we have,

$$k_x^2 = k_y^2 + k_z^2 \equiv k^2, \quad 3.40$$

where the total wavenumber, k , was now determined by the y- and z-direction wavenumbers.

3.3.3 The Perturbed Interface Equilibrium Relations

The surface perturbation created a disturbance that required integrating the Equation 3.34 across each interface to determine the perturbed pressure balance. This

required knowledge of the perturbed density, ρ_1 , which was found from Equations 3.35 and 3.36. The continuity equation with the incompressibility condition was therefore

$$\rho_1 + \xi_1 \cdot \nabla \rho_0 = 0. \quad 3.41$$

A critical simplifying observation was made; the fluid displacement in the x-direction was the most important quantity for the MRT instability. For example, the perturbed density became

$$\rho_1 = -\xi_{1x} \frac{\partial \rho_0}{\partial x}. \quad 3.42$$

Substituting Equation 3.42 into Equation 3.34 and integrating across the interface at $x = \alpha$ yielded

$$\int_{\alpha^-}^{\alpha^+} -\rho_0 \omega^2 \xi_{1x} dx = \int_{\alpha^-}^{\alpha^+} \left(\frac{\partial}{\partial x} [P_1 + P_{M1}] - g \xi_{1x} \frac{\partial \rho_0}{\partial x} \right) dx. \quad 3.43$$

Since the difference between α^- and α^+ was negligible the left hand side of Equation 3.43 was zero. Equation 3.43 became

$$0 = [P_1(\alpha^+) + P_{M1}(\alpha^+)] - [P_1(\alpha^-) + P_{M1}(\alpha^-)] - g \xi_{1\alpha} [\rho_{03} - \rho_{02}]. \quad 3.44$$

The integration for the interface at $x = \beta$ proceeded in the same manner and resulted in

$$0 = [P_1(\beta^+) + P_{M1}(\beta^+)] - [P_1(\beta^-) + P_{M1}(\beta^-)] - g \xi_{1\beta} [\rho_{02} - \rho_{01}]. \quad 3.45$$

Equations 3.44 and 3.45 described the perturbed pressure balance ultimately responsible for MRT instability development in terms of the displacement perturbations, $\xi_{1\alpha}$ and $\xi_{1\beta}$, at each interface. The unknown perturbed kinetic pressure, p_1 , and magnetic pressure, p_{m1} , immediately above and below each interface therefore needed to be determined in terms of $\xi_{1\alpha}$ and $\xi_{1\beta}$.

3.3.4 The Perturbed Kinetic Pressure

The perturbed kinetic pressure, P_1 , was determined by first coupling the y- and z-components of the perturbed displacement, ξ_{1y} and ξ_{1z} , to the x-component of the displacement, ξ_{1x} , using the components of Equation 3.34. Evaluating the $\mathbf{J} \times \mathbf{B}$ term in the equation of motion also required knowing the perturbed current density, \mathbf{J}_1 . Once this coupling was established, P_1 was easily obtained from the z-component of the equation of motion.

The perturbed current density, \mathbf{J}_1 , was coupled to the displacement vector, ξ_1 , by determining the perturbed magnetic field, \mathbf{B}_1 , and substituting this into Equation 3.38. Combining Equations 3.37 and 3.39 to find \mathbf{B}_1 yielded

$$\mathbf{B}_1 = \nabla \times (\xi_1 \times \mathbf{B}_0) = B_0 \nabla \times (\xi_1 \times \hat{\mathbf{z}}). \quad 3.46$$

Evaluating the cross products produced

$$\mathbf{B}_1 = B_0 \frac{\partial \xi_1}{\partial z} = -ik_z B_0 \xi_1, \quad 3.47$$

which is the “frozen-in” law in ideal MHD, which relates fluid displacement ξ_1 , to magnetic field perturbation \mathbf{B}_1 . The perturbed current density in terms of the perturbed displacement vector was therefore

$$\mathbf{J}_1 = (-ik_z) \frac{B_0}{\mu_0} \nabla \times \xi_1. \quad 3.48$$

The equilibrium density, ρ_0 , within each region was a constant, so the continuity and incompressibility conditions, Equations 3.35 and 3.36 respectively, forced ρ_1 within each region to be zero and removed the gravitational term from Equation 3.34. The components for the equation of motion within a given region were therefore

$$x: -\rho_0 \omega^2 \xi_{1x} = -\frac{\partial P_1}{\partial x} + ik_z \frac{B_0^2}{\mu_0} \left[\frac{\partial \xi_{1z}}{\partial x} + ik_z \xi_{1x} \right], \quad 3.49$$

$$y: -\rho_0 \omega^2 \xi_{1y} = ik_y P_1 - ik_z \frac{B_0^2}{\mu_0} [ik_y \xi_{1z} - ik_z \xi_{1y}], \quad 3.50$$

$$z: -\rho_0 \omega^2 \xi_{1z} = ik_z P_1. \quad 3.51$$

It was clear from Equation 3.51 in this form that P_1 could be expressed in terms of ξ_{1x} if the coupling between ξ_{1z} and ξ_{1x} was established. Multiplying Equation 3.50 by ik_z , Equation 3.51 by ik_y , and subtracting yielded

$$\rho_0 \omega^2 (ik_z \xi_{1y} - ik_y \xi_{1z}) = \frac{k_z^2 B_0^2}{\mu_0} (ik_z \xi_{1y} - ik_y \xi_{1z}). \quad 3.52$$

This equation was satisfied by two possibilities regarding the common factor $(ik_z \xi_{1y} - ik_y \xi_{1z})$. The first possibility, $(ik_z \xi_{1y} - ik_y \xi_{1z}) \neq 0$, allowed the common factor to divide out from each side of the equation, but this uninteresting result only produced the dispersion relation for an Alfvén wave. The second possibility, $(ik_z \xi_{1y} - ik_y \xi_{1z}) = 0$, was therefore pursued. This result coupled together the perturbed displacements in the y- and z-directions through

$$k_z \xi_{1y} = k_y \xi_{1z}. \quad 3.53$$

Equation 3.40 and 3.53 were then substituted into Equation 3.36 to couple the perturbed displacement in the x-direction to the perturbed displacements in the y- and z-directions

$$\xi_{1z} = \frac{-ik_z}{k^2} \frac{\partial \xi_{1x}}{\partial x}. \quad 3.54$$

Equations 3.53 and 3.54 therefore allowed the entire perturbed displacement vector to be expressed in terms of the x-component only. The perturbed kinetic pressure, P_1 , was

expressed in terms of the perturbed displacement in the x-direction by substituting Equation 3.54 into 3.51

$$P_1 = \frac{\rho_0 \omega^2}{k^2} \frac{\partial \xi_{1x}}{\partial x}. \quad 3.55$$

3.3.5 The Perturbed Magnetic Pressure

The perturbed magnetic pressure, P_{M1} , was expressed in terms of the perturbed displacement in the x-direction by first coupling the perturbed magnetic pressure to ξ_{1z} , and then relating the z-direction displacement perturbation to ξ_{1x} . First, the perturbed magnetic pressure was equal to the difference between the total magnetic pressure and the equilibrium magnetic pressure

$$P_{M1} = \frac{|\mathbf{B}_0 + \mathbf{B}_1|^2}{2\mu_0} - \frac{|\mathbf{B}_0|^2}{2\mu_0}, \quad 3.56$$

which linearized to

$$P_{M1} = \frac{B_0 B_{1z}}{\mu_0}, \quad 3.57$$

because the equilibrium magnetic field contained only a z-component. The z-component of the perturbed magnetic field was known from Equation 3.47 and the relation to ξ_{1x} was known from Equation 3.54. The resulting perturbed magnetic pressure in terms of ξ_{1x} was

$$P_{M1} = - \left(\frac{k_z}{k} \right)^2 \frac{B_0^2}{\mu_0} \frac{\partial \xi_{1x}}{\partial x}. \quad 3.58$$

3.3.6 The Perturbed Displacement

The perturbed kinetic and magnetic pressures were successfully expressed in terms of the x-component of the displacement ξ_{1x} ; however, the pressures required the

spatial derivative of ξ_{1x} , so the precise form for this quantity needed to be found as well. The equation of motion in x-direction, Equation 3.49, was not previously used, but the perturbed kinetic pressure given by Equation 3.55 allowed the pressure gradient term to be evaluated, and Equation 3.54 allowed all displacement vector components within Equation 3.49 to be converted to ξ_{1x} alone

$$-\rho_0 \omega^2 \xi_{1x} = -\frac{\omega^2 \rho_0}{k^2} \frac{\partial^2 \xi_{1x}}{\partial x^2} + \frac{B_0^2 k_z}{\mu_0} \left[\left(\frac{k_z}{k^2} \right) \frac{\partial^2 \xi_{1x}}{\partial x^2} - k_z \xi_{1x} \right]. \quad 3.59$$

Simplifying this equation yielded

$$\left(\omega^2 \rho_0 - \frac{B_0^2 k_z^2}{\mu_0} \right) \frac{\partial^2 \xi_{1x}}{\partial x^2} - \left(\omega^2 \rho_0 - \frac{B_0^2 k_z^2}{\mu_0} \right) k^2 \xi_{1x} = 0, \quad 3.60$$

$$\frac{\partial^2 \xi_{1x}}{\partial x^2} - k^2 \xi_{1x} = 0, \quad 3.61$$

where Equation 3.61 was a Laplace equation for ξ_{1x} that could be solved within each fluid region, given the proper boundary conditions, to determine the kinetic and magnetic pressures for that region.

3.3.7 Region 1 Pressures

The boundary conditions for Equation 3.61 for region 1 were that the perturbation $\xi_{1x}(x)$, had to equal $\xi_{1\beta}$ at the interface $x = \beta$, and that the perturbation had to decay to zero far below the interface. These boundary conditions were therefore

$$\xi_{1x}(x = \beta) = \xi_{1\beta}, \quad 3.62$$

$$\xi_{1x}(x \rightarrow -\infty) = 0, \quad 3.63$$

so the proper solution for Equation 3.61 for region 1 [Figure 3.2] was

$$\xi_{1x}(x) = \xi_{1\beta} e^{k(x-\beta)}, \quad x < \beta. \quad 3.64$$

Substituting the solution from Equation 3.64 into Equations 3.55 and 3.58

$$P_1(\beta^-) = \frac{\rho_{01}\omega^2}{k} \xi_{1\beta}, \quad 3.65$$

$$P_{M1}(\beta^-) = -\frac{k_z^2 B_0^2}{k\mu_0} \xi_{1\beta}, \quad 3.66$$

produced the perturbed kinetic and magnetic pressures just below the $x = \beta$ interface.

3.3.8 Region 2 Pressures

The boundary conditions for Equation 3.61 for region 2 were that the perturbation $\xi_{1x}(x)$, had to equal $\xi_{1\alpha}$ at the interface $x = \alpha$, and that the perturbation had to equal $\xi_{1\beta}$ at the interface $x = \beta$. These boundary conditions were therefore

$$\xi_{1x}(\alpha) = \xi_{1\alpha}, \quad 3.67$$

$$\xi_{1x}(\beta) = \xi_{1\beta}, \quad 3.68$$

so the proper solution for Equation 3.61 was

$$\xi_{1x}(x) = \xi_{1\alpha} \frac{\sinh(k(x-\beta))}{\sinh(k\Delta)} + \xi_{1\beta} \frac{\sinh(k(\alpha-x))}{\sinh(k\Delta)}, \quad \beta < x < \alpha. \quad 3.69$$

Substituting the solution from Equation 3.69 into Equations 3.55 and 3.58 for the interface at $x = \alpha$ yielded

$$P_1(\alpha^-) = \frac{\rho_{02}\omega^2}{k} \operatorname{csch}(k\Delta) [\xi_{1\alpha} \cosh(k\Delta) - \xi_{1\beta}], \quad 3.70$$

$$P_{M1}(\alpha^-) = -\frac{k_z^2 B_0^2}{k\mu_0} \operatorname{csch}(k\Delta) [\xi_{1\alpha} \cosh(k\Delta) - \xi_{1\beta}], \quad 3.71$$

and for the interface at $x = \beta$ yielded

$$P_1(\beta^+) = \frac{\rho_{02}\omega^2}{k} \operatorname{csch}(k\Delta) [\xi_{1\alpha} - \xi_{1\beta} \cosh(k\Delta)], \quad 3.72$$

$$P_{M1}(\beta^+) = -\frac{k_z^2 B_0^2}{k\mu_0} \operatorname{csch}(k\Delta) [\xi_{1\alpha} - \xi_{1\beta} \cosh(k\Delta)]. \quad 3.73$$

The perturbed kinetic and magnetic pressures for region 2 just below the interface at $x = \alpha$ and just above the interface at $x = \beta$ were therefore determined in terms of $\xi_{1\alpha}$ and $\xi_{1\beta}$.

3.3.9 Region 3 Pressures

The boundary conditions for Equation 3.61 for region 3 were that the perturbation $\xi_{1x}(x)$, had to equal $\xi_{1\alpha}$ at the interface $x = \alpha$, and that the perturbation had to decay to zero far above the interface. These boundary conditions were therefore

$$\xi_{1x}(\alpha) = \xi_{1\alpha}, \quad 3.74$$

$$\xi_{1x}(x \rightarrow \infty) = 0, \quad 3.75$$

so the proper solution for Equation 3.61 for region 3 [Figure 3.2] was

$$\xi_{1x}(x) = \xi_{1\alpha} e^{-k(x-\alpha)}, \quad x > \alpha. \quad 3.76$$

Substituting the solution from Equation 3.64 into Equations 3.55 and 3.58

$$P_1(\alpha^+) = -\frac{\rho_{03}\omega^2}{k} \xi_{1\alpha}, \quad 3.77$$

$$P_{M1}(\alpha^+) = \frac{k_z^2 B_0^2}{k\mu_0} \xi_{1\alpha}, \quad 3.78$$

produced the perturbed kinetic and magnetic pressures just above the $x = \alpha$ interface, also in terms of $\xi_{1\alpha}$ and $\xi_{1\beta}$.

3.3.10 The Dispersion Relation

The dispersion relation for the MRT instability was obtained by substituting the perturbed kinetic and magnetic pressures at each interface into the perturbed pressure balance equations. The equation for the $x = \alpha$ interface was found by substituting Equations 3.70, 3.71, 3.77, and 3.78 into Equation 3.44. The result was of the form

$$\xi_{1\alpha} = F_\alpha(\omega)\xi_{1\beta}, \quad 3.79$$

where the function $F_\alpha(\omega)$ coupled the perturbation on the $x = \beta$ interface, $\xi_{1\beta}$, to the perturbation on the $x = \alpha$ interface, $\xi_{1\alpha}$. A similar result for the $x = \beta$ interface was found by substituting Equations 3.65, 3.66, 3.72, and 3.73 into Equation 3.45 such that

$$\xi_{1\alpha} = F_\beta(\omega)\xi_{1\beta}. \quad 3.80$$

The functions $F_\alpha(\omega)$ and $F_\beta(\omega)$ were therefore

$$F_\alpha(\omega) = \left[\cosh(k\Delta) + \frac{\frac{k_z^2 B_{03}^2}{\mu_0} + gk(\rho_{02} - \rho_{03}) - \omega^2 \rho_{03}}{\frac{k_z^2 B_{02}^2}{\mu_0} - \omega^2 \rho_{02}} \sinh(k\Delta) \right]^{-1}, \quad 3.81$$

$$F_\beta(\omega) = \left[\cosh(k\Delta) + \frac{\frac{k_z^2 B_{01}^2}{\mu_0} + gk(\rho_{01} - \rho_{02}) - \omega^2 \rho_{01}}{\frac{k_z^2 B_{02}^2}{\mu_0} - \omega^2 \rho_{02}} \sinh(k\Delta) \right]. \quad 3.82$$

Equations 3.79 and 3.80 could not both be satisfied unless the condition

$$F_\alpha(\omega) = F_\beta(\omega), \quad 3.83$$

was true. This condition resulted in the general dispersion relation for the MRT instability for the 3 region planar geometry. The dispersion relation was a 4th order polynomial in ω of the form

$$A\omega^4 - B\omega^2 + C = 0, \quad 3.84$$

with coefficients

$$A = \rho_{02}^2 + \rho_{01}\rho_{03} + \rho_{02}(\rho_{01} + \rho_{03}) \coth(k\Delta) \quad 3.85$$

$$\begin{aligned} B = & \rho_{02}^2 (2k_z^2 v_{A2}^2) \\ & + \rho_{01}\rho_{02} [k_z^2 (v_{A1}^2 + v_{A2}^2) \coth(k\Delta) + kg(1 + \coth(k\Delta))] \\ & + \rho_{02}\rho_{03} [k_z^2 (v_{A2}^2 + v_{A3}^2) \coth(k\Delta) - kg(1 + \coth(k\Delta))] \\ & + \rho_{01}\rho_{03} [k_z^2 (v_{A1}^2 + v_{A3}^2)] \end{aligned} \quad 3.86$$

$$\begin{aligned}
C = & \rho_{02}^2 [k_z^2 v_{A2}^2 - (kg)^2] \\
& + \rho_{01} \rho_{02} [(k_z^2 v_{A1}^2 + kg)(k_z^2 v_{A2}^2 \coth(k\Delta) + kg)] \\
& + \rho_{02} \rho_{03} [(k_z^2 v_{A3}^2 - kg)(k_z^2 v_{A2}^2 \coth(k\Delta) - kg)] \\
& + \rho_{01} \rho_{03} [(k_z^2 v_{A1}^2 + kg)(k_z^2 v_{A3}^2 - kg)]
\end{aligned} \tag{3.87}$$

where v_{An} was the Alfvén speed, $v_{An} = B_{0n}/\sqrt{\mu_0 \rho_{0n}}$, in the region $n = 1, 2, 3$. The roots for Equation 3.84 were of the form

$$\omega^2 = \frac{1}{2A} (B \pm \sqrt{B^2 - 4AC}). \tag{3.88}$$

The energy principle for an ideal MHD description of the Rayleigh-Taylor instability required that ω^2 be real.⁵² Therefore, the condition $B^2 - 4AC \geq 0$ was always true. Additionally, the coefficient A was always positive from inspection. The stability conditions for the MRT instability were therefore

$$\text{if } B < 0 \rightarrow \text{always unstable,} \tag{3.89}$$

$$\text{if } C < 0 \rightarrow \text{always unstable,} \tag{3.90}$$

$$\text{if } C > 0 \text{ AND } B > 0 \rightarrow \text{stable.} \tag{3.91}$$

The eigenfrequencies, ω , from Equation 3.88 always occurred in positive and negative pairs such that

$$\omega_1 = -\omega_2, \tag{3.92}$$

$$\omega_3 = -\omega_4. \tag{3.93}$$

These eigenfrequencies were always either purely real or purely imaginary. A convention was adopted where ω_1 always referred to the most unstable mode.

3.4 Derivation of the Instability Feed-Through Effect

The instability coupling between interfaces was strongly suggested from Equations 3.79 and 3.80. In fact, this coupling allowed instability growing on an unstable

interface to generate instability amplitude onto a stable interface, the so called feed-through effect. The instability growth was an eigenmode for the entire system, including both interfaces, that evolved in time at the fastest temporal growth rate corresponding to the eigenfrequency ω_1 . Because the eigenmode of an instability means that the entire system evolves at the same growth rate, it followed that this evolution must have occurred even on a stable interface. While the growth rate was the same on either interface, the amplitude for a growing mode on each interface did not need to be the same. These amplitudes for a growing mode were coupled by Equation 3.79 or 3.80.

The rippled interface at $x = \beta$ contained a perturbation with amplitude $\xi_{1\beta}(t)$. This amplitude was expressed as a superposition of the four eigenmodes with eigenfrequencies $\omega_1, \omega_2, \omega_3$, and ω_4 ; and constant amplitudes p_1, p_2, p_3 , and p_4 respectively. Taking advantage of Equations 3.92 and 3.93, we write,

$$\xi_{1\beta}(t) = p_1 e^{i\omega_1 t} + p_2 e^{-i\omega_1 t} + p_3 e^{i\omega_3 t} + p_4 e^{-i\omega_3 t}. \quad 3.94$$

The p_n constants were determined from initial conditions. The equation for the $x = \alpha$ interface ripple, $\xi_{1\alpha}(t)$, was determined from Equation 3.94 by invoking Equation 3.80, the amplitude coupling between the $x = \beta$ interface to the $x = \alpha$ interface. The result was

$$\xi_{1\alpha}(t) = F_\beta(\omega_1)p_1 e^{i\omega_1 t} + F_\beta(\omega_1)p_2 e^{-i\omega_1 t} + F_\beta(\omega_3)p_3 e^{i\omega_3 t} + F_\beta(\omega_3)p_4 e^{-i\omega_3 t}, \quad 3.95$$

where the relations $F_\beta(\omega_1) = F_\beta(-\omega_1) = F_\alpha(\omega_1) = F_\alpha(-\omega_1)$; and from Equations 3.81 and 3.82, $F_\beta(\omega_3) = F_\beta(-\omega_3) = F_\alpha(\omega_3) = F_\alpha(-\omega_3)$ were used, and the p_n constants were determined from initial conditions. The initial conditions chosen to explore the feed-through effect were an initial perturbation, $\xi_{1\beta 0}$, on the interface $x = \beta$, zero initial perturbation on the interface $x = \alpha$, and zero initial velocity on either interface

$$\xi_{1\alpha}(0) = 0, \quad 3.96$$

$$\xi_{1\beta}(0) = \xi_{1\beta 0}, \quad 3.97$$

$$\frac{\partial \xi_{1\alpha}(0)}{\partial t} = 0, \quad 3.98$$

$$\frac{\partial \xi_{1\beta}(0)}{\partial t} = 0, \quad 3.99$$

which served to describe the feed-through effect from the unstable, perturbed interface at $x = \beta$ to the presumably stable, unperturbed interface at $x = \alpha$, assuming positive g pointing in the negative x -direction [Figure 3.2]. Equations 3.94 and 3.95, along with their corresponding time derivatives, provided 4 equations with boundary conditions described by Equations 3.96 - 3.99. This set of equations was used to solve for the 4 unknown p amplitude constants

$$p_1 = p_2 = \frac{\xi_{1\beta 0}}{2} \frac{F_\beta(\omega_3)}{F_\beta(\omega_3) - F_\beta(\omega_1)}, \quad 3.100$$

$$p_3 = p_4 = -\frac{\xi_{1\beta 0}}{2} \frac{F_\beta(\omega_1)}{F_\beta(\omega_3) - F_\beta(\omega_1)}, \quad 3.101$$

which were substituted into Equations 3.94 and 3.95 to yield

$$\xi_{1\beta}(t) = 2p_1 \cos(\omega_1 t) + 2p_3 \cos(\omega_3 t), \quad 3.102$$

$$\xi_{1\alpha}(t) = F_\beta(\omega_1) 2p_1 \cos(\omega_1 t) + F_\beta(\omega_3) 2p_3 \cos(\omega_3 t). \quad 3.103$$

Equations 3.102 and 3.103 only differed by the factor F_β for the eigenfrequencies ω_1 and ω_3 , therefore $F_\beta(\omega_1)$ was the feed-through factor for the eigenmode corresponding to $\omega_1 = -\omega_2$, and $F_\beta(\omega_3)$ was the feed-through factor for the eigenmode corresponding to $\omega_3 = -\omega_4$.

3.5 Specialized Cases

The generalized dispersion relation for the MRT instability was described by Equation 3.84 with coefficients A, B, and C given by Equations 3.85, 3.86, and 3.87 respectively. This general solution to the complete geometry shown in Figure 3.2 was applied to several more specialized cases in order to illustrate how these classical solutions were recovered from the general treatment.

3.5.1 Rayleigh Case: Two Regions of Infinite Depth

This geometry described the classical RT instability for 2 regions of infinite depth containing unmagnetized fluids sharing a single interface. The 3 region model was converted to a 2 region model by taking the limit where the region 2 thickness, Δ , went to infinity. The density and magnetic field in region 3 were set to zero, creating a 2 region geometry with a single interface at $x = \beta$. Alternatively, the region 2 thickness and density could have been set to zero to achieve the same result. The fluids were also unmagnetized so the magnetic fields in all regions were set to zero, leaving the classical RT instability geometry. The gravity vector, \mathbf{g} , was assumed to be pointing downward in the negative x-direction, and the negative sign already accounted for such that positive g represented the downward acceleration magnitude in the interface reference frame. For $\rho_{03} = 0, B_{01} = B_{02} = B_{03} = 0, \Delta \rightarrow \infty$ and arbitrary values for the remaining densities, the dispersion coefficients were

$$A = \rho_{02}^2 + \rho_{02}\rho_{01}, \quad 3.104$$

$$B = 2\rho_{01}\rho_{02}kg, \quad 3.105$$

$$C = -\rho_{02}^2(kg)^2 + \rho_{01}\rho_{02}(kg)^2. \quad 3.106$$

The dispersion relation was therefore

$$\omega^2 = \frac{\rho_1}{(\rho_2 + \rho_1)} kg \pm \frac{\rho_2}{(\rho_2 + \rho_1)} kg, \quad 3.107$$

with eigenfrequencies

$$\omega_1 = \sqrt{-\frac{(\rho_2 - \rho_1)}{(\rho_2 + \rho_1)} kg} = \sqrt{-A_t kg}, \quad 3.108$$

$$\omega_3 = \sqrt{kg}, \quad 3.109$$

where the dimensionless parameter

$$A_t = \frac{(\rho_2 - \rho_1)}{(\rho_2 + \rho_1)}, \quad 3.110$$

was the Atwood number for the fluid interface.

The stability for the ω_1 mode depended on the sign of the Atwood number. When the density in region 2 was higher than the density in region 1, corresponding to heavy fluid being supported by light fluid, A_t was positive, the eigenfrequency became imaginary, and the interface was therefore unstable. When the density in region 2 was lower than the density in region 1, corresponding to light fluid being supported by heavy fluid, A_t was negative, the eigenfrequency became real, and the interface was therefore stable. The ω_3 eigenfrequency was always real according to the energy principle and therefore was always a stable mode.

The growth rate $\gamma = \text{im}(\omega)$ for the RT instability was therefore

$$\gamma = \text{im}(\omega_1) = \text{im}(\sqrt{-A_t kg}), \quad 3.111$$

which completed the recovery of the classical Rayleigh scaling for the RT instability.

This geometry only contained a single interface, so the feed-through effect did not apply.

3.5.2 Taylor Case: Slab of Finite Thickness

Taylor's geometry differed from the Rayleigh case by containing 3 regions.⁴⁹

Regions 1 and 3 contained zero density and zero magnetic field. Region 2 contained an

unmagnetized fluid of finite density and thickness. For $\rho_{01} = \rho_{03} = 0, B_{01} = B_{02} =$

$B_{03} = 0$ and an arbitrary value for the remaining density, the dispersion coefficients were

$$A = \rho_{02}^2, \quad 3.112$$

$$B = 0, \quad 3.113$$

$$C = -\rho_{02}^2 (kg)^2. \quad 3.114$$

The dispersion relation was therefore

$$\omega^4 = kg, \quad 3.115$$

with eigenfrequencies

$$\omega_1 = \sqrt{-kg}, \quad 3.116$$

$$\omega_3 = \sqrt{kg}. \quad 3.117$$

The growth rate for the unstable mode was therefore

$$\gamma = im(\omega_1) = im(\sqrt{-kg}). \quad 3.118$$

Interestingly, Taylor's well known result was nearly identical to Rayleigh's result in that the finite slab thickness, Δ , never appeared in the dispersion relation. Additionally, a slab of finite thickness was always unstable, regardless of the direction for g , because one of the interfaces always contained the heavy region 2 fluid being "supported" by the surrounding vacuum. This geometry created an effective Atwood number, even though A_t never appeared explicitly in the dispersion relation, because the zero density for regions 1

and 3 resulted in $A_t = \pm 1$, thus only the sign for the interface for a given acceleration direction appeared in Equations 3.116 and 3.117.

The feed-through factors for Taylor's case were

$$F_\alpha(\omega) = \left[\cosh(k\Delta) - \left(\frac{gk}{\omega^2} \right) \sinh(k\Delta) \right]^{-1}, \quad 3.119$$

$$F_\beta(\omega) = \left[\cosh(k\Delta) - \left(\frac{-gk}{\omega^2} \right) \sinh(k\Delta) \right]. \quad 3.120$$

Substituting Equations 3.116 and 3.117 into Equation 3.120 yielded

$$F_\beta(\omega_1) = \cosh(k\Delta) - \sinh(k\Delta) = e^{-k\Delta}, \quad 3.121$$

$$F_\beta(\omega_3) = \cosh(k\Delta) + \sinh(k\Delta) = e^{k\Delta}, \quad 3.122$$

where the well known $\exp(-k\Delta)$ feed-through factor for the unstable mode was recovered.

3.5.3 Harris Case: Slab of Finite Thickness Supported by Magnetic Field

Harris' case was similar to Taylor's geometry except that region 1 also contained a magnetic field that supported the region 2 fluid slab.⁵⁰ For $\rho_{01} = \rho_{03} = 0, B_{02} = B_{03} = 0$ and arbitrary values for the remaining density and magnetic field, the dispersion coefficients were

$$A = \rho_{02}^2, \quad 3.123$$

$$B = \rho_{02}^2 k_z^2 v_{Al}^2 \coth(k\Delta), \quad 3.124$$

$$C = \rho_{02}^2 k g (k_z^2 v_{Al}^2 - k g), \quad 3.125$$

where $v_{Al}^2 = B_{01}^2 / (\mu_0 \rho_{02})$ was an interface Alfvén speed for the lower interface at $x = \beta$, where the mass was from the region 2 fluid, and the magnetic field was from region 1.

The resulting dispersion relation was

$$\omega^4 - k_z^2 v_{Al}^2 \coth(k\Delta) \omega^2 + kg(k_z^2 v_{Al}^2 - kg) = 0. \quad 3.126$$

Harris wrote his equilibrium pressure balance as [see Equation 3.25]

$$\frac{B_{01}^2}{2\mu_0} = \rho_{02} g \Delta \quad 3.127$$

when expressed in terms of the Figure 3.2 geometry and in SI units. Substituting Equation 3.127 into Equation 3.126 explicitly recovered Harris' dispersion relation [Equation 26 of Harris⁵⁰]

$$\omega^4 - k_z^2 2\Delta g \coth(k\Delta) \omega^2 - (kg)^2 \left(1 - \frac{k_z^2}{k} 2\Delta\right) = 0. \quad 3.128$$

The eigenfrequencies were therefore

$$\omega_1 = \sqrt{\frac{1}{2} k_z^2 v_{Al}^2 \coth(k\Delta) - \frac{1}{2} \sqrt{(k_z^2 v_{Al}^2 \coth(k\Delta))^2 - 4kg(-kg + k_z^2 v_{Al}^2)}}, \quad 3.129$$

$$\omega_3 = \sqrt{\frac{1}{2} k_z^2 v_{Al}^2 \coth(k\Delta) + \frac{1}{2} \sqrt{(k_z^2 v_{Al}^2 \coth(k\Delta))^2 - 4kg(-kg + k_z^2 v_{Al}^2)}}, \quad 3.130$$

and the growth rate for the most unstable mode was therefore

$$\gamma = im(\omega_1) = im\left(\sqrt{\frac{1}{2} k_z^2 v_{Al}^2 \coth(k\Delta) - \frac{1}{2} \sqrt{(k_z^2 v_{Al}^2 \coth(k\Delta))^2 - 4kg(-kg + k_z^2 v_{Al}^2)}}\right). \quad 3.131$$

The feed-through factors were

$$F_\alpha(\omega) = \left[\cosh(k\Delta) - \frac{gk}{\omega^2} \sinh(k\Delta)\right]^{-1}, \quad 3.132$$

$$F_\beta(\omega) = \left[\cosh(k\Delta) - \frac{-gk + k_z^2 v_{Al}^2}{\omega^2} \sinh(k\Delta)\right]. \quad 3.133$$

Substituting Equations 3.129 and 3.130 into Equation 3.133 yielded the feed-through factors for each eigenfrequency

$$F_{\beta}(\omega_1) = \left[\cosh(k\Delta) - \frac{(-gk + k_z^2 v_{A1}^2) \sinh(k\Delta)}{\frac{1}{2}k_z^2 v_{A1}^2 \coth(k\Delta) - \frac{1}{2}\sqrt{(k_z^2 v_{A1}^2 \coth(k\Delta))^2 - 4kg(-kg + k_z^2 v_{A1}^2)}} \right], \quad 3.134$$

$$F_{\beta}(\omega_3) = \left[\cosh(k\Delta) - \frac{(-gk + k_z^2 v_{A1}^2) \sinh(k\Delta)}{\frac{1}{2}k_z^2 v_{A1}^2 \coth(k\Delta) + \frac{1}{2}\sqrt{(k_z^2 v_{A1}^2 \coth(k\Delta))^2 - 4kg(-kg + k_z^2 v_{A1}^2)}} \right]. \quad 3.135$$

Since the dispersion coefficient B was always positive, the instability condition given by Equation 3.90 demanded $C < 0$ to be true. This instability condition became

$$\frac{k_z^2 B_{01}^2}{\mu_0 \rho_{02}} < kg, \quad g > 0, \quad 3.136$$

or, using Equation 3.127, the instability condition becomes

$$k > k_z^2(2\Delta), \quad 3.137$$

which is Harris' Equation 36.⁵⁰ For negative g , meaning $\mathbf{g} = g\hat{\mathbf{x}}$ in this geometry, the interface at $x = \alpha$ was always unstable regardless of the magnetic field in region 1. For positive g there was a stabilization mechanism for the instability on the $x = \beta$ interface due to the magnetic field. This important result was stated by Harris, but with Equation 3.127 substituted into the condition, which masked the magnetic field as the underlying mechanism. With the magnetic field explicitly included, the stabilizing role of the field became clearer. Unstable surface ripples with wavenumber parallel to the magnetic field, k_z , experienced a stabilizing effect because such ripples required the field lines to bend, whereas surface ripples with wavenumber perpendicular to the field, k_y , did not require the field lines to bend and therefore experienced no effect from the magnetic field. The asymmetry in the feed-through factors also resulted from the magnetic field only existing in region 1 and not in region 3. The presence of magnetic field in the region opposite to the unstable interface reduced the feed-through factor to that interface. The magnetic

field therefore served to reduce instability growth with spatial frequency components parallel to the field, both by reducing the instability growth rate at the unstable interface due to magnetic field at that interface; and by reducing the feed-through factor due to magnetic field at the opposite interface. This stabilization mechanism and anisotropy due to the magnetic field were key differences between the RT and MRT instabilities.

3.5.4 Experimental Case: Slab of Finite Thickness Surrounded by Magnetic Field

The geometry used by Harris was extended to include a magnetic field in region 3. Therefore this geometry represented the experimental geometry modeled by Figure 3.1. For $\rho_{01} = \rho_{03} = 0, B_{02} = 0$ and arbitrary values for the remaining density and magnetic fields, the dispersion coefficients were

$$A = \rho_{02}^2, \quad 3.138$$

$$B = \rho_{02}^2 k_z^2 \coth(k\Delta)(v_{Al}^2 + v_{Au}^2), \quad 3.139$$

$$C = \rho_{02}^2 [(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)]. \quad 3.140$$

The dispersion relation was therefore

$$\omega^4 - k_z^2 \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) \omega^2 + [(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)], \quad 3.141$$

with eigenfrequencies

$$\omega_1 = \sqrt{\frac{1}{2} \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) - \frac{1}{2} \sqrt{[\coth(k\Delta)(v_{Al}^2 + v_{Au}^2)]^2 - 4(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)}}, \quad 3.142$$

$$\omega_3 = \sqrt{\frac{1}{2} \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) + \frac{1}{2} \sqrt{[\coth(k\Delta)(v_{Al}^2 + v_{Au}^2)]^2 - 4(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)}}. \quad 3.143$$

The growth rate for the most unstable mode was therefore

$$\gamma = \text{im}(\omega_1) = \text{im} \left(\sqrt{\frac{1}{2} \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) - \frac{1}{2} \sqrt{[\coth(k\Delta)(v_{Al}^2 + v_{Au}^2)]^2 - 4(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)}} \right), \quad 3.144$$

and the feed-through factors were

$$F_{\alpha}(\omega) = \left[\cosh(k\Delta) - \frac{k_z^2 v_{Au}^2 + gk}{\omega^2} \sinh(k\Delta) \right]^{-1}, \quad 3.145$$

$$F_{\beta}(\omega) = \left[\cosh(k\Delta) - \frac{k_z^2 v_{Al}^2 + gk}{\omega^2} \sinh(k\Delta) \right]. \quad 3.146$$

Substituting Equations 3.142 and 3.143 into Equation 3.146 yielded the feed-through factors for each eigenfrequency

$$F_{\beta}(\omega_1) = \left[\cosh(k\Delta) - \frac{(gk + k_z^2 v_{Al}^2) \sinh(k\Delta)}{\frac{1}{2} \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) - \frac{1}{2} \sqrt{[\coth(k\Delta)(v_{Al}^2 + v_{Au}^2)]^2 - 4(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)}}} \right], \quad 3.147$$

$$F_{\beta}(\omega_3) = \left[\cosh(k\Delta) - \frac{(gk + k_z^2 v_{Al}^2) \sinh(k\Delta)}{\frac{1}{2} \coth(k\Delta)(v_{Al}^2 + v_{Au}^2) + \frac{1}{2} \sqrt{[\coth(k\Delta)(v_{Al}^2 + v_{Au}^2)]^2 - 4(-kg + k_z^2 v_{Al}^2)(kg + k_z^2 v_{Au}^2)}}} \right]. \quad 3.148$$

The dispersion coefficient B was always positive, so the instability condition given by Equations 3.90 required $C < 0$ to hold for instability to occur. This instability condition became

$$\frac{k_z^2 B_{01}^2}{\mu_0 \rho_{02}} < kg, \quad g > 0, \quad 3.149$$

$$\frac{k_z^2 B_{03}^2}{\mu_0 \rho_{02}} < -kg, \quad g < 0. \quad 3.150$$

The instability conditions were identical to the Harris case except that magnetic field was present at each interface, so the stabilization mechanism was present at the unstable interface regardless of the acceleration direction. The feed-through factors were also reduced for either sign of g because magnetic field was always present at the opposite interface. Again, the magnetic field only served to stabilize MRT modes with spatial frequency components parallel to the field since such modes required the magnetic field

lines to bend. Modes with spatial frequency components perpendicular to the magnetic field were not stabilized by this effect.

One special case worth noting for this geometry involved the limiting case where region 2 was very thick compared to the instability wavelength, $k\Delta \gg 1$. This effectively decoupled the interfaces. Setting the density and magnetic field in region 3 equal to zero resulted in a single interface geometry with growth rate

$$\gamma = im(\omega_1) = im\left(\sqrt{-kg + k_z^2 v_{Al}^2}\right). \quad 3.151$$

The instability condition for this case was the same as Equation 3.149. This result was shown by Chandrasekhar.⁵¹ Here the growth rate stabilization due to the magnetic field was very clear without the additional complication from a second interface.

CHAPTER 4

MAGNETO-RAYLEIGH-TAYLOR EXPERIMENTAL CONFIGURATION

The magneto-Rayleigh-Taylor experiments were carried out at the MAIZE pulsed power facility at the University of Michigan's Plasma, Pulsed Power, and Microwave Laboratory.⁵ This facility was constructed by a large collaborative effort between primary graduate students J.C. Zier, M.R. Gomez, D.M. French, and S.G. Patel; support staff M. Perreault and A. Thurtell; under the supervision and guidance of Professors R.M. Gilgenbach and Y.Y. Lau. The pulsed power machine used to produce the large currents required to drive the MRT instability utilized linear transformer driver (LTD) architecture.^{6,7,53} These currents were transmitted through an evacuated magnetically insulated transmission line (MITL) from the generator output electrodes to the load.⁵⁴ The load itself consisted of a cartridge designed to transport and to secure in place the fragile 400 nm thick Al foil targets. The drive current from the LTD was conducted through the foil, turning it into metallic plasma susceptible to the magneto-Rayleigh-Taylor instability. MRT instability growth on this plasma was measured using a laser backlighter diagnostic to image the unstable plasma and to record its dynamic evolution. The experimental shot procedure is provided in Appendix F.

4.1 The MAIZE Pulsed Power Driver

The MAIZE facility consisted of a single module shown in Figure 4.1 (a) with the top metallic cover, internal plastic insulation, and insulating oil removed to expose the

internal brick subunits. This LTD module contained 40 bricks (2 capacitors and a switch) that, when combined, were capable of delivering 1 MA of current at 100 kV into a 0.1 Ω matched impedance load with a 100 ns risetime.⁶ Auxiliary facility components, such as the HV charging circuit, compressed gas system, triggering circuitry, and computer control system are described in Sections 4.1.3 - 4.1.6 respectively.

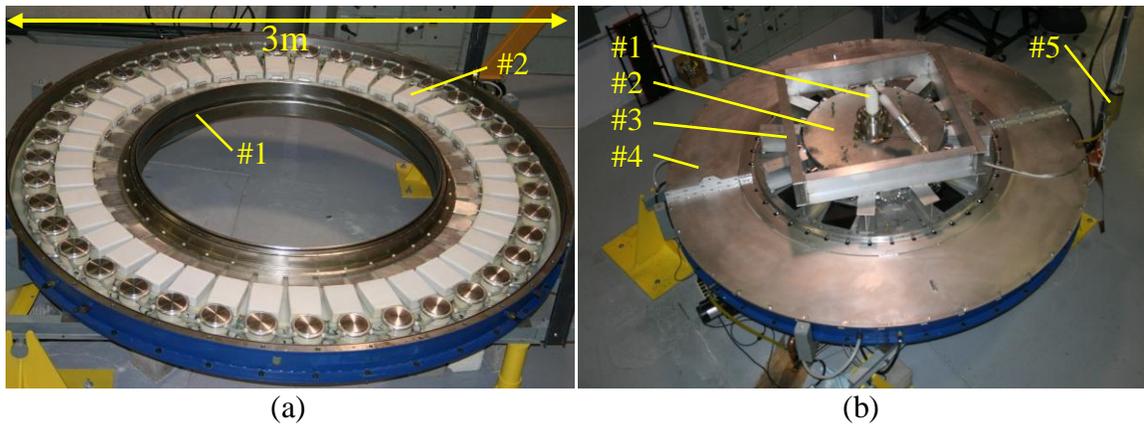


Figure 4.1 (a) An image of the MAIZE LTD with the vacuum and oil flanges, top HV insulator, and top core removed to expose the 40 bricks. KEY: 1) AK gap, 2) Brick subunits (b) The same device with the central vacuum flanges, experimental chamber, and oil section cover installed. KEY: 1) High voltage feed-through and Northstar HV probe. This voltage diagnostic was not available for the MRT shots because the feed-through could not be attached to the cathode hardware without blocking the laser diagnostic, 2) Experimental vacuum chamber bolted to anode vacuum flange, 3) I-beam supports to reduce deflection on the anode flange due to vacuum pressure, 4) Removable oil section cover, 5) Oil section header.

For the specific module shown in Figure 4.1, the bricks were housed inside the 3 m diameter, 22 cm thick, metallic pillbox-shaped toroid casing and sandwiched between large plastic insulators to guarantee sufficient standoff between the high voltage electrodes and the casing. With the top cover installed, the casing itself was oil tight so that transformer oil could be pumped inside to surround the bricks with additional insulation. The AK gap insulator between the output electrodes formed an oil/vacuum interface seal. Vacuum flanges were mated to the module along with the necessary vacuum system components to evacuate and monitor the vacuum section, such as a rough vacuum rotary pump, a cryogenic high vacuum pump, a thermocouple gauge, and ion

gauges. The vacuum pumps were connected through the lower vacuum flange below the experiment. The experimental chamber and lid completed the vacuum section as shown in Figure 4.1 (b). The vacuum system was capable of bringing the chamber from atmospheric pressures to ~ 50 μTorr in an hour, and to an ultimate pressure of ~ 5 μTorr if left on high vacuum overnight.

4.1.1 LTD Brick Configuration

The 40 brick subunits used to form the module shown in Figure 4.1 each contained two double-ended, 40 nF, General Atomics capacitors in series separated by a spark gap switch able to withstand 200 kV and 25 kA. The total inductance and resistance for each brick was 240 nH and 600 m Ω respectively; 115 nH and 60 m Ω in the switch; 20 nH and 540 m Ω in the two capacitors combined; and 105 nH in the remaining buses.^{6,7,55,53} The matched impedance for each brick was 4 Ω . The total module contained 800 nF of capacitance, 6 nH of inductance, and a 0.1 Ω matched impedance. The switches used were developed at the Institute of High Current Electronics in Tomsk, Russia. These switches contained six gaps that utilized small needles on the cathode side of each intermediate electrode to generate a coronal discharge to resistively grade the potential across the gaps.⁷ The capacitors were charged on the switch side electrode. As oriented in the laboratory, the top capacitors were charged to $-V_c$ and the bottom capacitors were charged to $+V_c$, where V_c was the desired charge voltage. All the top switch electrodes from each brick were connected together through ~ 1 k Ω liquid charging resistors made from 23 g/L copper sulfate solution, forming a ring through which the entire top half of the module was charged through a single location. The bottom switch electrodes were similarly connected to form the bottom charging ring. The

details of the charging circuit are provided in Section 4.1.3. Inlet and outlet gas fittings allowed the switches to be filled and purged with dry synthetic air, the details of which are provided in Section 4.1.4. The switches were triggered by a high voltage pulse delivered to the midplane of each brick switch through $\sim 1\text{ k}\Omega$ liquid trigger resistors made from 30 g/L potassium bromide solution. Additional trigger system details are provided in Section 4.1.5.

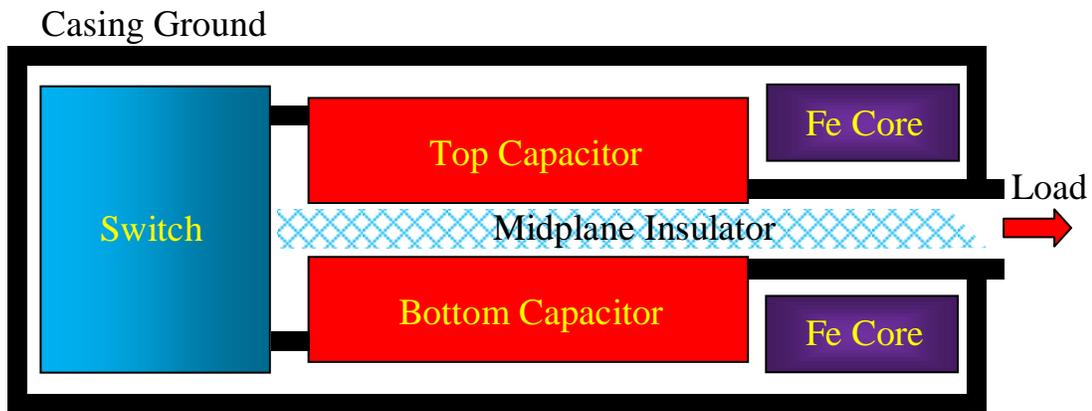


Figure 4.2 A cross-sectional cartoon of the brick geometry for a LTD. The insulator slant shown above is for a cathode output electrode on the bottom.

The output bus from the top capacitor in each brick was connected to a single anode, and the output bus from the bottom capacitor in each brick was similarly connected to a single cathode. The top output electrode became the anode even though the top capacitors were negatively charged because closing the spark gap switch forced both the negative top and positive bottom charging electrodes to be at the same potential. The potential difference therefore appeared across the capacitors onto the output electrodes with the reverse polarity from the charging voltages, hence the top output electrode became the machine anode, and the bottom output electrode became the machine cathode.

4.1.2 Magnetic Cores and Premagnetization

The ferromagnetic cores, shown in Figure 4.4, served to make the direct path to case ground a much more inductive route for the drive current pulse than the path through the load. The drive pulse contained a large dI/dt on the order of 10^{13} A/s, so the current division between the parallel paths was dominated by the inductive impedance rather than the resistive impedance, which allowed the ferromagnetic cores to force the drive current through the load. After the drive pulse, any stray charge on the capacitors or other slow dI/dt currents were not affected by the core inductances and therefore saw the nearly zero resistance path to the casing, which passively grounded the output side of the capacitors. Since the cores were ferromagnetic, they experienced a hysteresis effect due to the alignment of magnetic dipoles inside the material with the imposed magnetic field. The magnetic field from the drive current induced a magnetization vector, \mathbf{M} , inside the core material such that

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}). \quad 4.1$$

Once no further magnetization was possible, the magnetic core saturated and no longer isolated the casing ground path. The maximum magnetization swing occurred when the magnetic dipoles inside the ferromagnetic material were all pointed in the completely opposite direction before the drive current began. The drive current then had to flip the maximum number of magnetic dipoles before the core saturated. This process, referred to as premagnetization, was accomplished using a LC premagnetization pulse generator (PPG) charged to 3 kV with a 20 μH inductance, 40 μF capacitance, and a diode across the capacitor to eliminate ringing. The PPG produced a 1.5 kA pulse with a 40 μs

risetime that passed around the cores in the opposite direction as the drive pulse in order to reverse the magnetization direction.⁵⁵ An image of the PPG is shown in Figure 4.3.



Figure 4.3 A photograph of the premagnetization pulse generator.

The PPG brought the core hysteresis magnetization to negative saturation, which was allowed to relax to the -1.2 T remanence. This relaxation after the premagnetization pulse, referred to as passive premagnetization, was necessary because the current path around the cores was still in parallel with the load. Ideally, the PPG would have been triggered during the shot sequence just before the main LTD pulse, referred to as active premagnetization, in order to utilize the full core magnetization swing, but the PPG pulse would have still delivered enough current through the fragile foil load to destroy it before the main drive occurred. The cores were therefore passively premagnetized immediately after a shot when the load no longer existed in order to reset the core magnetization

polarization for the next shot. For the ferromagnetic-tape cores used inside the LTD, the saturation magnetic field, B_s , was 2 T at a 30 kA/m auxiliary magnetic field; the remnant magnetic field, B_r , was 1.2 T; and the coercive force, H_c , was 30 A/m.^{6,55} The hysteresis loop shape was approximated as

$$B_{upper}(H) = B_s \frac{H+H_c}{|H+H_c|+H_c\left(\frac{B_s}{B_r}-1\right)} + \mu_0 H, \quad 4.2$$

$$B_{lower}(H) = B_s \frac{H-H_c}{|H-H_c|+H_c\left(\frac{B_s}{B_r}-1\right)} + \mu_0 H, \quad 4.3$$

where B_{upper} and B_{lower} represented the upper and lower curves of the loop, the same model incorporated by the LTSPICE⁵⁶ code used to model the LTD circuit in Chapter 5.

The hysteresis curve for the LTD cores, based on Equations 4.2 and 4.3, is shown in Figure 4.5. The volt-second integral, VS, for a given voltage drive $U(t)$, was related to the magnetic field change inside the ferromagnetic cores through

$$VS = \int U(t)dt = S * \Delta B, \quad 4.4$$

for a given core magnetic cross-sectional area, S . The LTD cores had a 64 cm² total cross-sectional area for the iron tape, so the 3.2 T field swing from the negative remnant magnetic field to the positive saturation magnetic field resulted in a volt-second integral of 20.5 mVs.⁵⁵ This quantity represented the maximum total magnetic flux the cores would withstand before magnetic saturation. After saturation, the cores would no longer force the main drive current to flow through the load and would instead allow any remaining energy to dump into the grounded casing. Prior to saturation, the cores were found to act like a nearly constant resistance in parallel with the variable core inductance. This resistance, due to eddy current formation inside the 500 nΩ*m resistivity

ferromagnetic iron, was approximately 1.5Ω and dominated the core behavior prior to saturation.⁵⁵

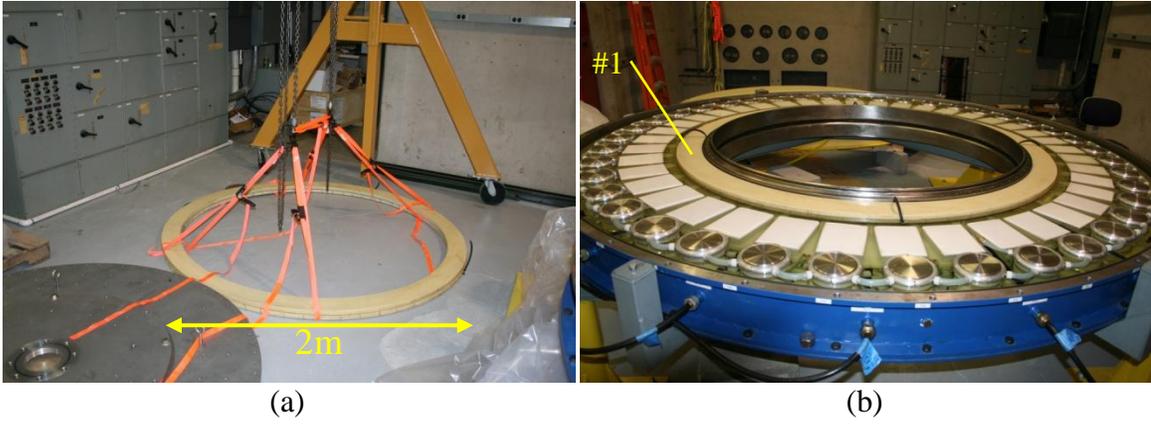


Figure 4.4 (a) An image of the top ferromagnetic core during installation. A crane was needed to move the 2 m diameter insulated iron ring into the machine. (b) An image of the top core after being installed into the LTD module. KEY: 1) The top ferromagnetic core installed in the machine.

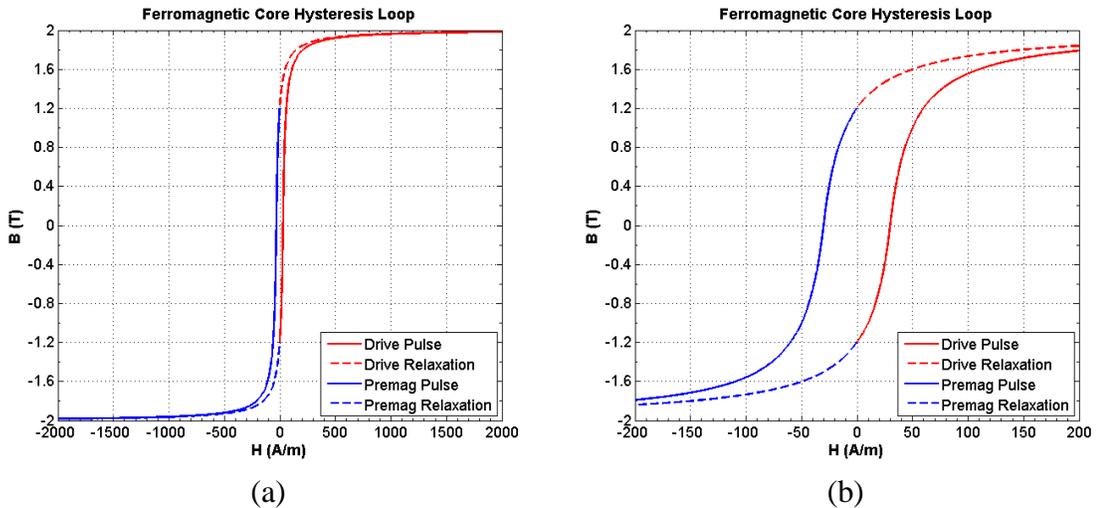


Figure 4.5 (a) A plot of the approximate LTD core hysteresis loop with the drive pulse (red) assumed to produce a positive field swing, and the premagnetization pulse (blue) producing a negative field swing, resulting in counter-clockwise motion around the hysteresis loop. The relaxation paths for each direction were shown as dashed lines. (b) The same hysteresis loop where the x-axis was expanded to accentuate the loop details around the remnant magnetization points.

4.1.3 High Voltage Charging System

The LTD was charged using two Glassman high voltage supplies, a negative supply for charging the top capacitor ring, and a positive supply for charging the bottom capacitor ring and the high voltage trigger. Each supply was rated to 125 kV and 30 mA,

although the complete system was only rated for use up to 100 kV for either polarity. A 20 mA charging current was typically used to charge the LTD module. Images of the high voltage supply and control rack are shown in Figure 4.6

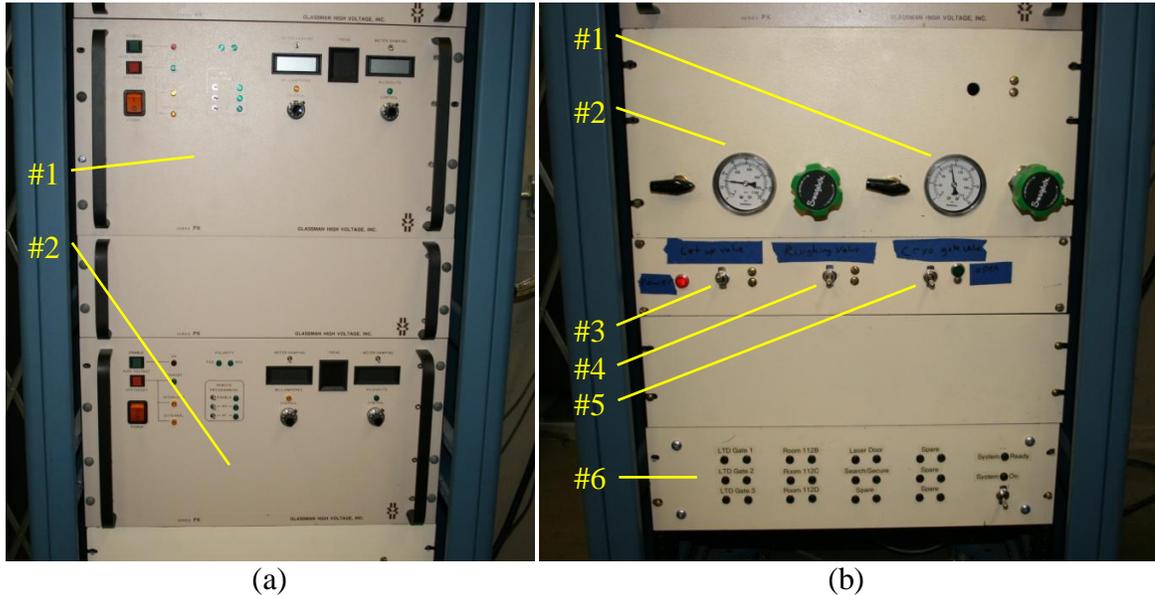


Figure 4.6 (a) An image of the high voltage supplies used to charge the LTD module. KEY: 1) Positive Glassman high voltage supply, 2) Negative Glassman high voltage supply. (b) The spark gap switch pressure controls, vacuum system controls, and interlock indicator panel for the LTD module. KEY: 1) Pressure controls for trigger switch, 2) Pressure controls for module switches, 3) Letup valve control, 4) Roughing pump valve control, 5) Cryogenic pump gate valve control, 6) Area interlock indicator panel.

The high voltage circuit connections were manipulated using three high voltage Ross-relays as shown in Figure 4.7. A Ross-relay connected each high voltage supply and protection circuit to its respective capacitor ring during the charging process when energized, and disconnected the power supply from the machine immediately before firing the shot, or in the event of a power failure. The third relay connected the capacitor rings and high voltage trigger to ground through 2 k Ω discharge resistors. This connection to ground was the default position for the relay and allowed any charge on the capacitors to drain to ground through the resistors in the event of an aborted shot, after a successful shot, or in the event of a power failure. Energizing the ground relay

disconnected the capacitors from ground and allowed the capacitors to be charged by the high voltage supplies. The voltages were monitored using voltage dividers on both the positive and negative polarity circuits on the capacitor bank side of the power supply connection relays. These voltage monitors allowed the potential on the capacitors to be monitored even after the power supply relays were disconnected immediately prior to a shot. The calibrations for the resistive dividers are provided in Appendix E.

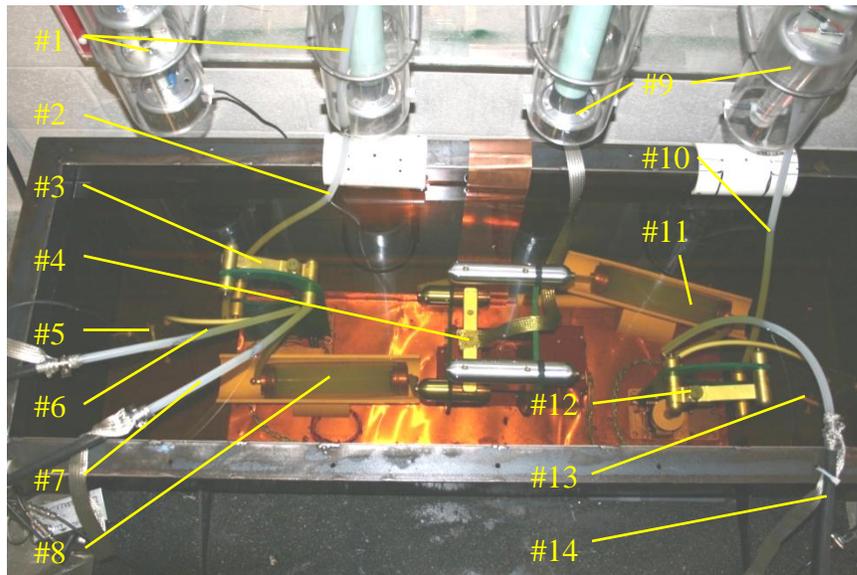


Figure 4.7 A photograph of the high voltage relay system used to control the MAIZE LTD module. The high voltage connections were placed in a grounded metallic tank filled with insulating oil to prevent arcing. **KEY:** 1) Positive protection circuits, 2) Positive feed from Glassman high voltage supply, 3) Positive power supply relay, 4) Ground relay, 5) Positive resistive divider voltage monitor, 6) Positive trigger charging feed, 7) Positive LTD module charging feed, 8) Positive dump resistor between capacitors and ground relay, 9) Negative protection circuits, 10) Negative feed from Glassman high voltage supply, 11) Negative dump resistor between capacitors and ground relay, 12) Negative power supply relay, 13) Negative resistive divider voltage monitor, 14) Negative LTD module charging feed.

The relay sequence for a shot began with the power supplies disconnected, and the ground relay connected. To begin the charging process, the power supply relays received a *PS Connect* signal from the control system, described in detail in Section 4.1.6, that energized them and connected the high voltage supplies to the capacitors. At

the same time, the ground relay received an *Unabort* signal that energized this relay and disconnected the capacitors from ground. The ground sides of the discharge resistors became floating nodes, removing the discharge resistors from the circuit while the ground relay was lifted. The high voltage supplies then charged the capacitors. Once capacitors were fully charged the *PS Connect* signal was stopped, causing the power supply relays to disengage and to disconnect the high voltage supplies from the capacitors. The machine was then triggered to perform the experiment, after which the *Unabort* signal was stopped in order to drop the ground relay. This reconnected the discharge resistors to ground, restoring the resistive ground path for the capacitors and returning the machine to a safe, discharged state. In the event of an abort during the shot sequence, the power supply relays were disconnected and the ground relay dropped to drain off whatever charge was present on the capacitors.

4.1.4 Pressure System

The LTD module required two compressed air systems. One system used compressed air from a shop air compressor unit to actuate solenoid valves. The second system used dry synthetic air from a compressed gas cylinder bank to pressurize and purge the spark gap switches inside the LTD. These systems were constructed by M.R. Gomez, J.C. Zier, and D.M. French. This purified air was required to prevent contaminants from degrading the switch performance. A diagram of the pressure system is shown in Figure 4.8.

The shop air system was set to 80 PSIG and was connected to all the vacuum system valves. This allowed the four solenoid valves to be electronically triggered and physically actuated by the gas pressure. Three solenoid valves activated by manual

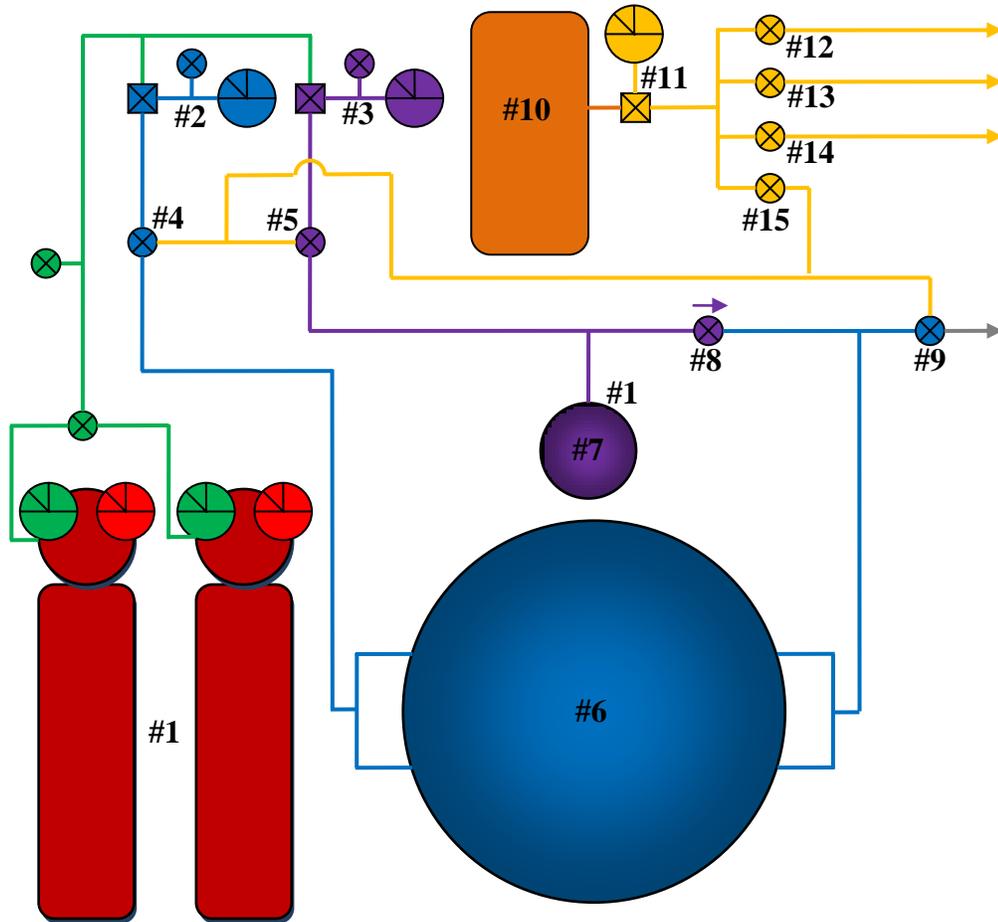


Figure 4.8 A diagram showing the pressure system for the MAIZE facility's LTD module. **KEY:** 1) Compressed synthetic air cylinders, 2) LTD module switch system regulator, gauge, and vent, 3) LTD trigger switch system regulator, gauge, and vent, 4) Normally open pneumatic inlet valve for LTD module switches, 5) Normally open pneumatic inlet valve for LTD trigger switch, 6) LTD module switches, 7) LTD trigger switch, 8) 1 PSI check valve, 9) Normally closed pneumatic exhaust valve to vacuum ballast tank and roughing pump (not shown), 10) Shop air compressor, 11) Shop air regulator and gauge, 12) Solenoid valve to LTD chamber letup pneumatic valve (not shown), 13) Solenoid valve to LTD chamber roughing pump pneumatic valve (not shown), 14) Solenoid valve to LTD chamber cryogenic pump gate valve (not shown), 15) Solenoid valve for LTD switch purge cycle.

switches controlled the roughing valve, which opened the roughing pump to the vacuum chamber; the cryogenic valve, which opened the gate valve to connect the cryogenic pump to the chamber; and the letup valve, which vented the chamber to atmosphere or to an alternative letup gas reservoir, such as dry nitrogen or helium. These controls are shown in Figure 4.6. The 4th solenoid valve activated the switch purge sequence for the synthetic air system when it received the *Purge Switches* control signal. This solenoid

valve supplied pressure to three pneumatic valves, one normally open valve on the input to the LTD module spark gap switches, one normally open valve on the input to the trigger spark gap switch, and one normally closed valve on the exhaust from both spark gap switch systems. The *Purge Switches* signal therefore closed off the synthetic air supply inputs to both the LTD module switches and the trigger switch, and opened the exhaust valve from both systems. The exhaust was connected to a ballast vacuum chamber and to a roughing vacuum pump. The switches were purged to vacuum in order to quickly remove used air from the gas system. When the *Purge Switches* signal was lowered, the solenoid valve closed, causing the exhaust valve to close and the input valves to open again. Fresh gas from the dry synthetic air cylinders then entered both spark gap switch systems. The LTD module switch system contained two inlets and two exhausts in parallel with each other to improve airflow through the module switches. The trigger switch contained a single line that served as both the inlet and the exhaust for the switch. The synthetic air supply was set to 60 PSIG on the cylinder regulators themselves, which provided a large enough source pressure to achieve an acceptable gas refill rate, and to ensure that adequate pressure was available for any required spark gap switch pressure setting. The actual spark gap switch pressures were set by two additional regulators on the inputs to the module switches and to the trigger switch. The control panel for these pressure systems is shown in Figure 4.6. These pressure settings, shown in Table 1, were charge voltage dependent and were empirically determined from machine operation.

The appropriate module and trigger pressures were set prior to each shot for the chosen charging voltage. The trigger pressure for 100 kV operation was not empirically

Table 1 Operating pressures for the LTD module and trigger spark gap switches for various charge voltages.

Charge Voltage (kV)	LTD Module Pressure (PSIG)	LTD Trigger Pressure (PSIG)
70	28	11.5
80	34	13
90	40	16
100	46	N/A

known because the entire module was not operated beyond ± 90 kV, but was expected to be roughly 18 PSIG. The pressures used for the dynamic foil loads were increased slightly from the standard set points. For the 70 kV charge voltage used for the foil shots, the actual pressures were 28.5 PSIG and 12 PSIG. This pressure increase was chosen to nearly eliminate the risk of module prefires, which would have destroyed the foil load without producing experimental data. In principle, the spark gap switch triggering jitter should have increased due to the increased pressure, but any jitter from this source was not noticeable. Since the module switch pressure was always larger than the trigger switch pressure, a cost saving simplification was made to the pressure system; a 1 PSIG check valve was placed at the exhaust for the trigger gas line before it connected with the module exhaust. This valve prevented any gas flow from the module side to the trigger side, but allowed both systems to share the same exhaust. This simplification removed the need for a second exhaust valve. A minor consequence of this simplification was that the trigger switch pressure could not be set higher than the module switch pressure, so the module switches had to refill above the intended trigger switch pressure after a gas purge before the trigger pressure could be properly set to its final value.

The gas lines inside the module oil section needed to be reconfigured from the original design. Since the gas fittings on the spark gap switches inside the module were approximately 135° apart and the gas tubing bending radius made direct connection

between neighboring switches very difficult, the original switch orientation for these lines used an alternating pattern where the gas fittings were placed radially outward and then inward from one switch to the next. This configuration, shown in Figure 4.9, required S-shaped gas lines that had to be thermally stressed in order to form the sharp bends. These lines were susceptible to cracking, which would potentially allow insulating oil into the spark gap switches and potentially cripple the LTD module.

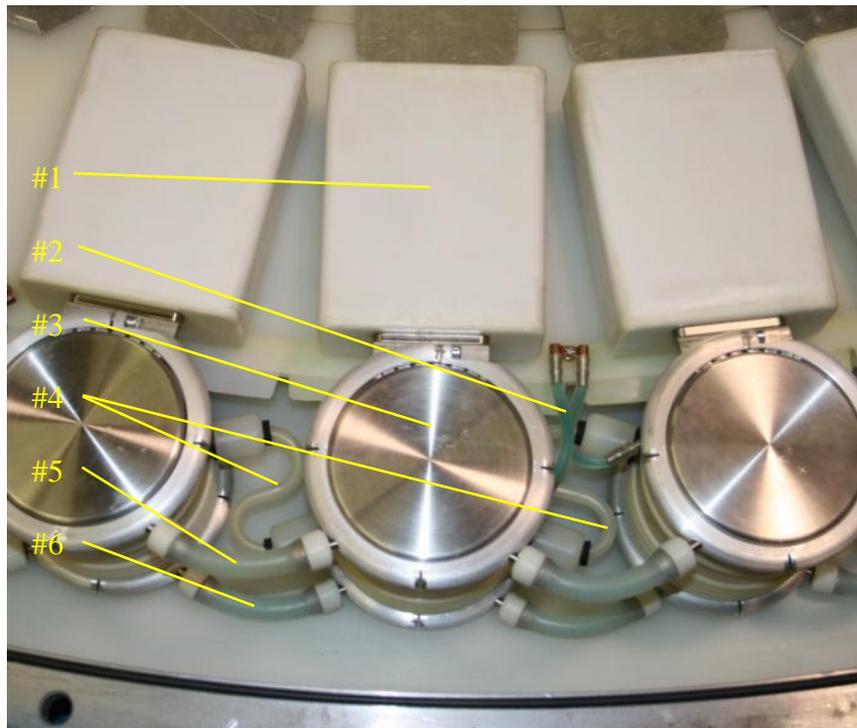


Figure 4.9 An image of the original gas line configuration. The fragile S-shaped gas lines were required due to the alternating gas fitting configuration from switch to switch. **KEY:** 1) Topside capacitors, 2) Trigger resistors, 3) Spark gap switches, 4) S-shaped gas tubing, 5) Negative charging ring resistors, 6) Positive charging ring resistors.

The improved configuration, shown in Figure 4.10 (a), oriented all the gas fittings outward. To solve the tubing bending radius problem each switch was connected to its next nearest neighbor. This created four parallel paths from the two gas inlets through the module switches to the two exhaust ports. The paths were connected at the inlet and exhaust ports as shown in Figure 4.10 (b).



Figure 4.10 (a) An image of the improved gas line configuration for the LTD module spark gap switch pressure system. The switches were connected between markers 1-1 and 2-2. The liquid resistors were also reconstructed using oil-resistant tubing, further improving the reliability of the LTD module and resulting in the green color. (b) The configuration used to crosslink the parallel gas line paths at the inlet and exhaust ports. The switch gas fittings 1-4 were connected to the inlet port via 3 T-shaped connectors.

These improvements reduced the stress on the tubing and fitting seals, which greatly enhanced the reliability of the LTD module.

4.1.5 LTD Module Trigger System

The triggering sequence for the LTD module is shown in Figure 4.11. This chain was required to progress from a software *Fire* signal to the high voltage pulse used to break down the module spark gap switches. The *Fire* signal from the Labview system, described in Section 4.1.6, triggered an eight channel Berkeley Nucleonics 575 delay generator. The delay generator was responsible for all triggers that occurred on a sub-millisecond timescale during the shot. The delay generator could not directly trigger the thyatron unit because this circuit contained a nonstandard 35Ω trigger termination that required at least a 30 V trigger signal. Additionally, the thyatron unit, shown in Figure 4.12, required a $10 \mu\text{s}$ long pulse, presumably to deliver enough current to trigger the internal components. This problem was overcome by using a DEI PVX-4140. This piece of equipment accepted an externally supplied bias up to $\pm 3.5 \text{ kV}$ and generated this

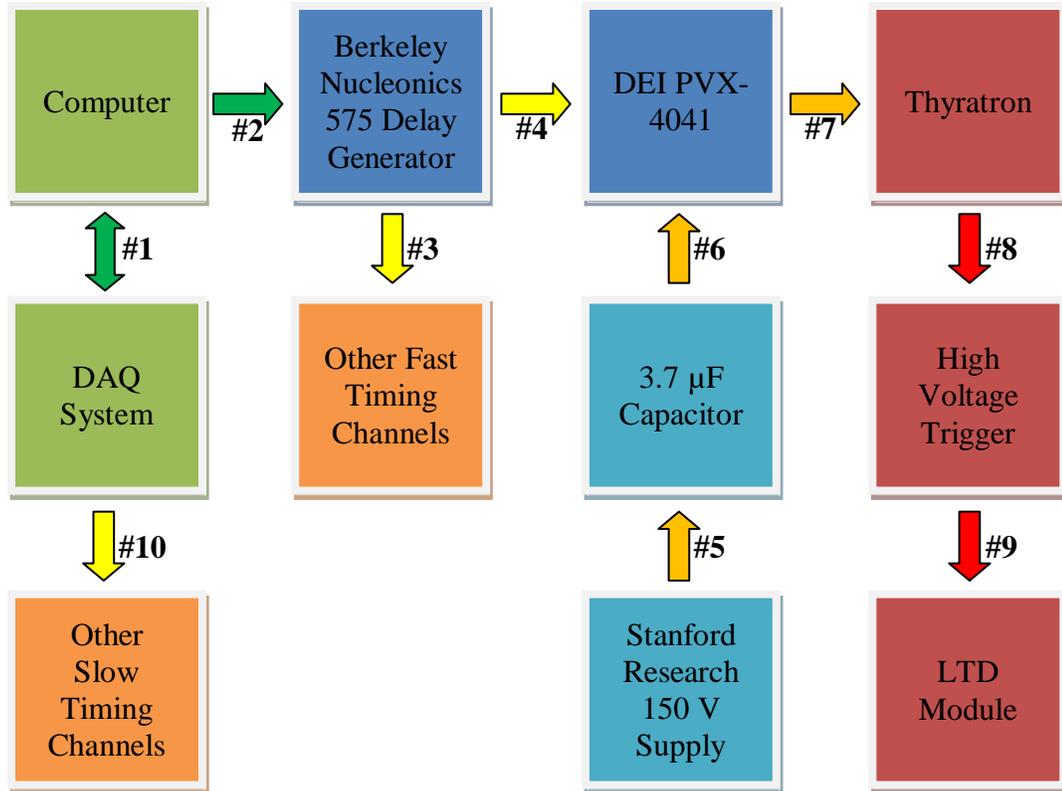


Figure 4.11 The triggering progression for the MAIZE LTD module. **KEY:** 1) Bi-directional software communication via USB to determine when to trigger the experiment, 2) Software trigger via RS-232 connection, 3) TTL triggers to additional diagnostic systems, 4) 10 μ s TTL gate trigger, 5) +150 VDC bias to capacitance, 6) +150 VDC bias backed by storage capacitance for sufficient current output, 7) ~50 V trigger to thyatron from capacitor discharge, 8) Grounding of intermediate spark gap plane to break down high voltage trigger switch, 9) High voltage trigger to each LTD module quadrant to trigger the machine, 10) Voltage and shorting commands to additional diagnostic systems.

supplied bias when an external TTL gate was sent to the unit. The output pulse was the same length as the TTL gate with a 24 ns rise and fall time. One of the delay generator channels was therefore used to send a 10 μ s TTL gate pulse, and the output from the DEI was sent to the thyatron trigger input. The external bias was a 3.7 μ F capacitance constructed from four parallel 1 μ F capacitors rated for up to 200 V. These capacitors were charged using a Stanford Research PS 310 high voltage supply set to 150 V. The additional voltage headroom above 30 V was required because the DEI unit was originally designed to drive high impedance loads and contained an internal



Figure 4.12 (a) A side view of the thyatron unit showing the trigger input and the sync output connections. (b) A front view of the thyatron unit. When the circuit became too warm the thyatron did not trigger reliably. The additional cooling fan was added to prevent this heat buildup, which resulted in reliable triggering.

impedance roughly double the termination impedance, resulting in a voltage division. The pulse that actually reached the thyatron was approximately 50 V and reliably triggered the thyatron circuit.

The thyatron triggered the high voltage side of the system. The high voltage trigger consisted of a 20 nF General Atomics capacitor and a single spark gap switch configured like a single capacitor LTD brick. These components were placed inside an oil-filled metal cylinder as shown in Figure 4.16. The high voltage trigger was energized using the same positive polarity charging voltage as was used for the module bricks. This trigger switch contained 5 resistively divided gaps with the thyatron in parallel with the final gap. The thyatron therefore had 1/5 of the positive polarity module charge voltage across it. The system was triggered by the thyatron switch, which closed to double the potential across the next gap inside the trigger spark gap switch. This gap broke down

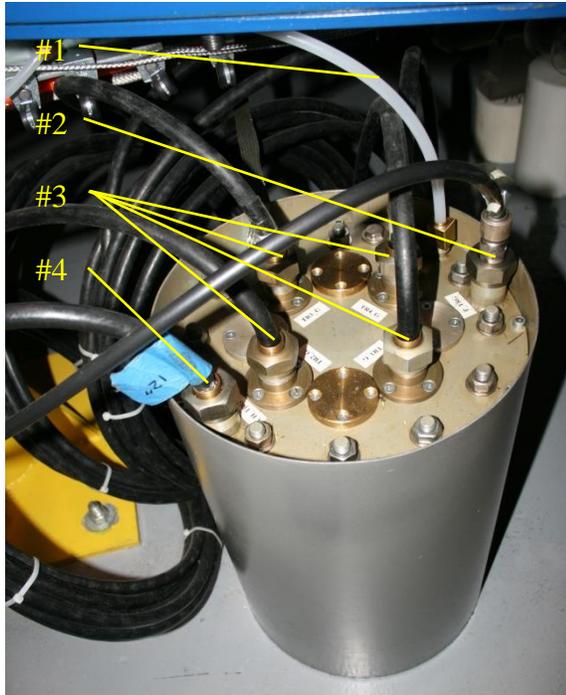


Figure 4.13 The high voltage trigger for the LTD module. **KEY:** 1) Trigger spark gap switch gas line, 2) Cable to thyatron switch, 3) High voltage trigger outputs for the LTD module, 4) Positive charging cable.

and caused a cascading breakdown across the entire switch, allowing the capacitor to send a negative polarity trigger pulse, equal in magnitude to the trigger charge voltage, to the four high voltage trigger cables with a 30 ns risetime. These 65Ω high voltage trigger cables entered the module symmetrically and attached to a single azimuthal trigger wire loop inside the midplane insulator.⁶ The $\sim 1 \text{ k}\Omega$ trigger resistors attached to this wire and to the trigger plane of each module brick. The potential spike caused by the trigger pulse then broke down the module spark gap switches and discharged the machine into the load.

4.1.6 Experimental Control and Data Acquisition Systems

The MAIZE facility was controlled using three primary components: a computer running a Labview control program that accepted all the configuration settings, controlled

the system hardware, and stored the experimental data; a data acquisition (DAQ) system that served as an interface between the Labview software and much of the experimental hardware; and a relay control system responsible for the switch purge cycle, high voltage relay operation, interlocks, and abort logic. This system was enclosed inside a copper mesh screen room with copper pipes leading to shielded feed-through panels around the facility. This metallic enclosure acted as a Faraday shield to protect the electronic components from the hostile electromagnetic pulse generated from the experiment during a shot. A diagram of the control and data acquisition system is shown in Figure 4.14. Detailed settings used for these components during the MRT foil shot series are provided in Appendix F.

Experimental data was stored on the MAIZE control computer in addition to being backed up onto the laboratory file server. The Labview program running on the MAIZE computer was developed by A. Thurtell with mission guidance, hardware, and debugging support from J.C. Zier. This program contained a graphical user interface (GUI) that served as a flexible framework for controlling power supplies, oscilloscopes, delay generators, and relays. The GUI contained a tabbed structure where information for the shot control sequence, delay generators, oscilloscopes, and power supplies were each on a separate tab. A screenshot of the GUI is provided in Figure 4.15. The desired settings for component channels were configured here, as were the power supply charging controls. The shot sequence was a list of bi-directional instructions between various hardware components, DAQ channels, or power supplies to control the hardware for preshot tests, laser backlighter operations, and for the shot itself. This system allowed the entire list of operations required to fire the backlighter laser, LTD module, or both to

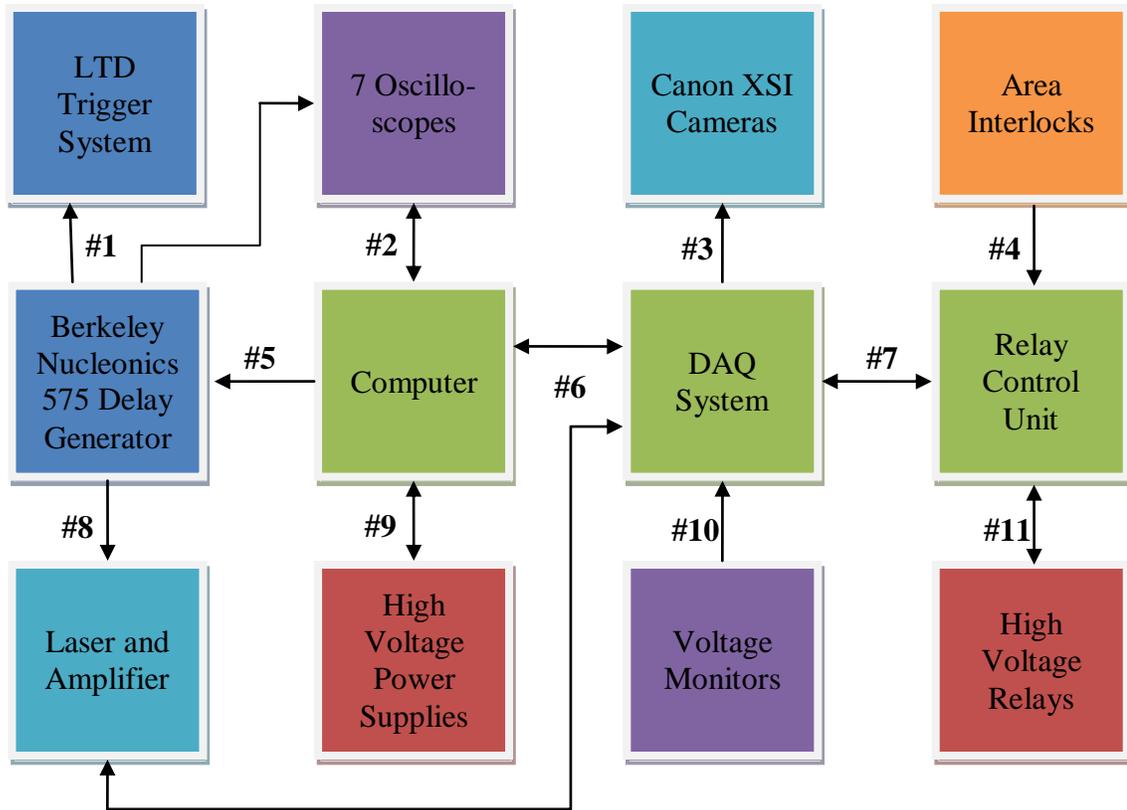


Figure 4.14 A diagram showing the experimental control and data acquisition components. KEY: 1) TTL *Fire* and oscilloscope command triggers, 2) GPIB data link, 3) Shorting relay for shutters, 4) Monitor for area interlocks, 5) RS232 data link for programming and *Fire* signal, 6) Computer ↔ DAQ USB data link. DAQ→Laser signals were the *Laser Charge* and *Laser Dump* amplifier controls. Laser→DAQ signal was *Laser Monitor* for the amplifier capacitor voltage, 7) DAQ→Relay Control Unit signals were *PS Connect*, *Unabort*, and *Purge Switches*. Relay Control Unit → DAQ signals were *Ground Monitor*, *PS Connect Monitor Pos*, *PS Connect Monitor Neg*, *Area Interlock*, *Equipment Interlock*, and *Ready* signals, 8) TTL trigger to laser Q-switch, 9) RS232 data links for the positive and negative Glassman high voltage supplies, 10) *Voltage Pos* and *Voltage Neg* signals for monitoring LTD capacitor voltages, 11) 120 VAC interface to Ross Relays.

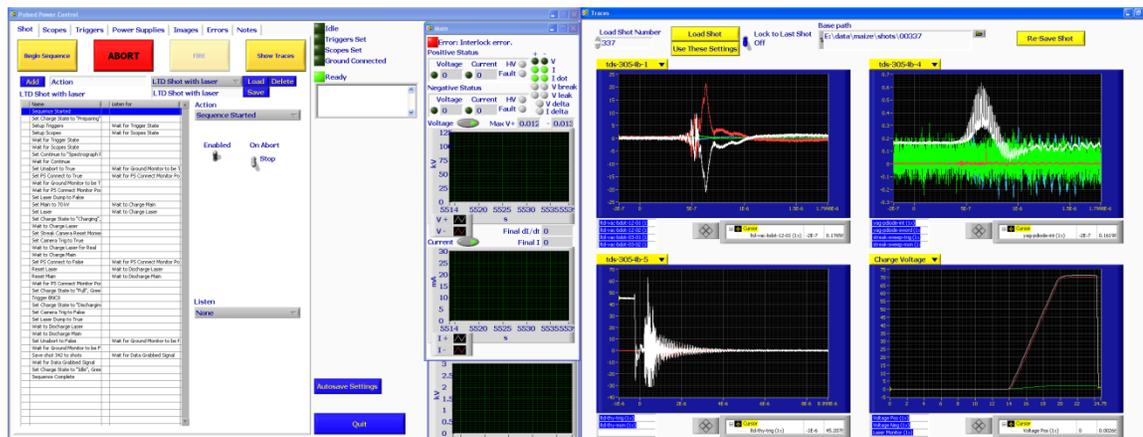


Figure 4.15 A screenshot of the Labview control program GUI. The power supplies and interlocks were not set when this particular screenshot was taken, so the power supply status indicated an error.

be executed by a single mouse click. The details for operating the Labview control program are documented in the user's manual.⁵⁷

The second component was the DAQ unit, designed by J.C. Zier. This system was required for converting software commands from Labview into analog or digital voltages for controlling experimental components that were not directly linked to the control computer via RS232 or GPIB connections. Images of the DAQ unit I/O panel and the internal components are shown in Figure 4.16. The DAQ also accepted analog and digital inputs from the experiment that were passed to Labview. This rack-mounted control box contained a NI-6229 DAQ system and supporting circuitry with capacity for: 1) 4 analog output channels via BNC connectors, 2) 16 analog input channels via BNC connectors, 3) 8 timed digital programmable function interface (PFI) input/output channels via BNC connectors, 4) 8 buffered digital input channels via screw terminals, 5) 4 buffered 5 V digital output channels via screw terminals, 6) 5 buffered 8.3 V digital output channels via screw terminals, and 7) 2 shorting relay output channels via BNC connectors. In addition, the DAQ system contained a 25 pin D-subminiature connector with 12 buffered 5 V digital input channels, 7 buffered 8.3 V digital output channels, 1 ground connection, and 5 unused pins. The DAQ communicated with the Labview computer via a USB connection. The buffered channels were passed through a series of 3-state octal buffer chips. These chips were needed to protect the sensitive and expensive NI-6229 from any voltage transients present on the I/O cables from RF sources during the shot, as well as to act as operational amplifiers to bolster the current and voltage capabilities of the NI-6229 for driving external components. The details of the DAQ I/O are provided in Appendix E. This system was designed to handle the existing requirements for the MAIZE facility, as

well as to have significant headroom for future facility expansion. The DAQ was used to monitor the LTD module voltages through analog inputs with 10 kHz lowpass RC filters, as well as the laser amplifier voltage through an analog input with a 3.3 kHz lowpass RC filter. The DAQ was also used to control the laser backlighter amplifier system, as well as the camera shutters. The primary control signals; however, passed between the DAQ system and the relay control system via the 25 pin connection. The *PS Connect*, *Unabort*, and *Purge Switches* signals were sent to the relay box; and the *Ground Monitor*, *PS Connect Monitor Pos*, *PS Connect Monitor Neg*, *Area Interlock*, *Equipment Interlock*, and *Ready* signals were sent to the DAQ. The DAQ output signals to the relay control system were commands to actuate the high voltage relays or the pressure system purge solenoid. Input signals from the relay box were all confirmation signals that the high voltage relays and interlocks were properly set as commanded.

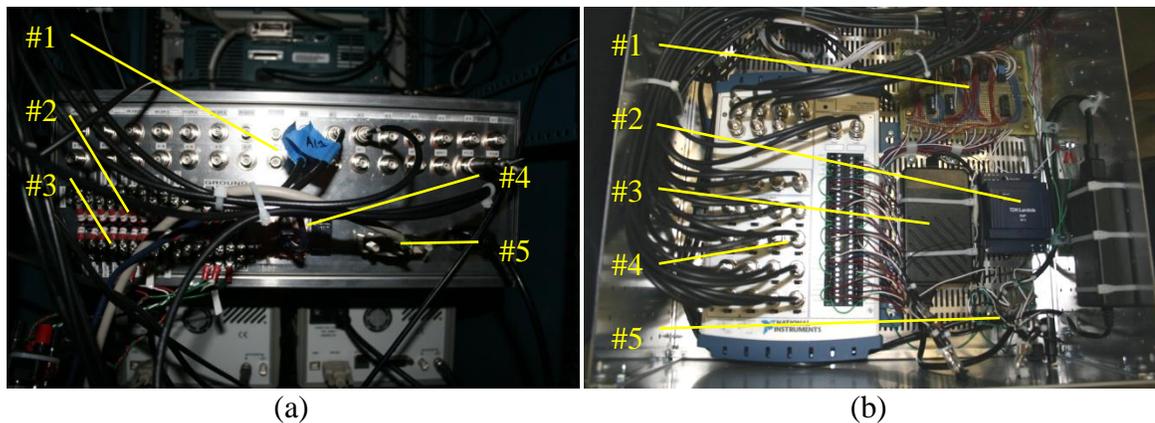


Figure 4.16 (a) An image of the installed DAQ unit's rear I/O panel showing the 25 pin connector, two screw terminals, numerous BNC connectors, and USB port. **KEY:** 1) BNC connection ports, 2) Screw terminal 1, 3) Screw terminal 2, 4) USB connector, 5) 25 pin D-sub connector. (b) An image of the DAQ unit's internal components showing the NI-6229, power supplies, and buffer chip circuitry. **KEY:** 1) Buffer chip circuitry, 2) 5 VDC supply, 3) 8.3 VDC supply, 4) NI-6229 DAQ, 5) 120 VAC bus.

The relay control box, shown in Figure 4.17, contained the 120 VAC relays used to control the high voltage relays and the gas purge system. This unit was configured to

operate in either the default computer controlled mode, or a manual mode where the relay cycle could be controlled from the front panel. The manual abort button for the experiment, as well as the power key, was located on the front panel. The area interlock system shown in Figure 4.6 for the LTD module perimeter, as well as the high voltage equipment interlocks, were tied into the relay control box such that an interlock fault



Figure 4.17 An image of the relay control unit's front panel showing the status indicator lights, switch purge button, manual operation controls, abort button, and power key.

would automatically trip the abort button and shut down the system. The pinout for the rear screw terminals is provided in Appendix E. These screw terminal connections could not be easily shielded, so the control cables were passed through 10 kHz RC lowpass filters at the entrance to the Faraday perimeter to screen out RF noise induced on the cables.

4.2 Power Flow and Load Hardware

The main current drive from the LTD module AK gap electrodes was delivered to the foil load through a magnetically insulated transmission line (MITL), where the self-generated magnetic field from the high current density inhibited emitted electrons from the cathode from crossing the transmission line gap. The foil load itself was held in place inside the load cartridge at the output from this transmission line. This cartridge

mechanically supported the foil during the loading procedure outside the machine, during transport from the loading area to the experimental chamber, and during the experiment itself.

4.2.1 Transmission Line

The MITL consisted of three major sections: a coaxial feed, a radial feed, and a double-sided planar riser feed. A computer aided design (CAD) rendering of the MITL hardware is shown in Figure 4.18. The coaxial and radial sections were designed by M.R. Gomez and the riser section was designed by J.C. Zier and D.M. French. The inductance for the MITL sections needed to be minimized due to the 0.1Ω matched impedance for the LTD module in order to efficiently transfer power from the generator to the load, and due to the limited voltage available since the MAIZE facility used only a single module.

For a given current loop

$$V = -\frac{\partial\Phi}{\partial t} = -\Lambda\frac{\partial I}{\partial t} - I\frac{\partial\Lambda}{\partial t}, \quad 4.5$$

where V is the loop voltage, Φ is the magnetic flux through the loop, Λ is the inductance of the loop, and I is the current around the loop. The second term was zero for the MITL plates because they did not move, so the required voltage to support a given current rise was directly proportional to the loop inductance. The LTD module was a high current, fast risetime generator with a dI/dt on the order of 10^{13} A/s, but the single module only produced 10^5 V into a matched load, so inductances on the order of 10 nH were required. Otherwise the generator voltage would not have been able to support the fast current rise and the current through the load would have been reduced.

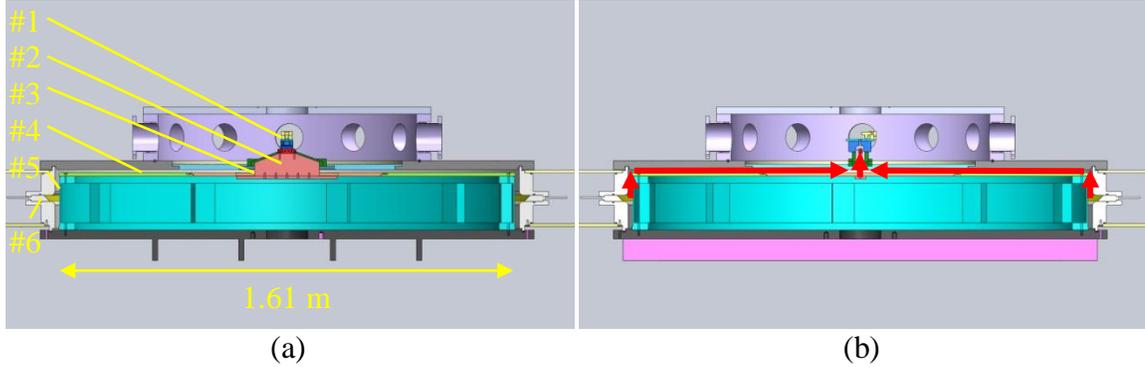


Figure 4.18 A CAD rendering of the MAIZE LTD module MITL sliced along (a) the North-South machine axis, and (b) along the East-West machine axis with the power flow shown in red. **KEY:** 1) Load cartridge, 2) MITL riser section, 3) MITL transition section, 4) MITL radial section, 5) MITL coaxial section, 6) LTD module AK gap.

The coaxial section AK gap spacing was 1.5 cm at an 80.59 cm inner radius. This large gap spacing was permissible because the large radius for this section reduced the current density and the resulting magnetic field, so the added inductance was trivial and provided an extra safety margin during installation and operation. The inductance for a volume V filled with magnetic field was calculated using either

$$\Lambda = \frac{1}{\mu_0 I^2} \int B^2 dV, \quad 4.6$$

$$\Lambda = \frac{1}{I} \int \mathbf{B} \cdot d\mathbf{A}. \quad 4.7$$

Applying Equation 4.7 to the coaxial section geometry of height h yielded

$$\Lambda = \frac{\mu_0 h}{2\pi} \ln \left(\frac{R_{out}}{R_{in}} \right). \quad 4.8$$

For $h = 10.64$ cm, $R_{out} = 82.09$ cm, and $R_{in} = 80.59$ cm the inductance was 0.39 nH, which proved to be a small contribution to the total MITL inductance as expected from the reduced current density and magnetic field at this radius.

The anode for the radial section of the MITL was unfortunately also a vacuum flange. Deflection of this flange and compression of o-ring seals due to atmospheric

pressure acting on this surface made setting the AK gap nontrivial. The gap spacing was 1.2 cm prior to any compression. The 2 mm of o-ring compression brought the nominal gap spacing to 1 cm, but the vacuum plate deflection measured at a 43.18 cm radius was approximately an additional 2 mm, suggesting that the MITL spacing may have approached 0.5 cm near the riser section. The exact profile for the deflection was not known, so Equation 4.7 was applied using 1 cm for the AK gap spacing, g , to produce an upper bound on the inductance

$$\Lambda = \frac{\mu_0 g}{2\pi} \ln \left(\frac{R_{out}}{R_{in}} \right). \quad 4.9$$

This calculation was performed for the radial section before the transition region to the double planar riser section broke the cylindrical symmetry of the transmission line, from an 80.59 cm outer radius to a 16.71 cm inner radius. The upper estimate for this inductance was 3.15 nH. The inductance for the transition region could not be calculated exactly, so a rough approximation was used to estimate the inductance for this region. The current through the transition region was assumed to only flow perpendicular to the long dimension, X , of the riser section slot of width, Y . The geometry therefore reduced to a limiting case with two parallel rectangular feeds of length equal to the long dimension of the riser section, and width for each rectangle assumed to extend to the radius of the long dimension. Applying Equation 4.7 to this approximate geometry yielded

$$\Lambda \approx \frac{\mu_0 g}{2X} \left(\frac{X}{2} - Y \right). \quad 4.10$$

For the parameter values $g = 1$ cm, $X = 33.43$ cm, and $Y = 1.65$ cm the approximate inductance was 2.83 nH.

The riser section contained a double-sided planar feed that sloped upward to the load cartridge mounting point. The bottom of the feed was rectangular and the top of the feed was a trapezoid that sloped toward the load mounting point. Equation 4.6 was applied to this geometry neglecting the fringing magnetic field at the plate edges. For a rectangular section height, h_r , a trapezoidal section height, h_t , a long trapezoidal base length, X_b , and a short trapezoidal top length, X_t , the inductance was

$$\Lambda = \frac{\mu_0 g}{2X_b} \left(h_r + \left(\frac{h_t}{X_t/X_b - 1} \ln \left(\frac{X_t}{X_b} \right) \right) \right). \quad 4.11$$

For the parameter values $g = 1$ cm, $h_b = 5.08$ cm, $h_t = 1.89$ cm, $X_b = 23.50$ cm, and $X_t = 6.16$ cm the inductance was 2.28 nH for the riser section. The total estimated transmission line inductance was therefore 8.65 nH for all three sections. This inductance was approaching the 10 nH level without taking into account the foil load itself, which was expected to contain several nanohenries of inductance due to the high current density and magnetic field present in the load region. The total impedance for the load and transmission line was therefore expected to be mildly overmatched to the generator. This result, combined with estimates shown in Chapter 5, forced the LTD module to be charged to 70% of capacity in order to protect the components from the higher voltages expected from the overmatched coupling.

4.2.2 Experimental Chamber Configuration

The experimental chamber contained 12 vacuum ports with 11.43 cm diameter openings evenly spaced around the chamber wall. A CAD rendering and an annotated photograph of the experimental chamber region are shown in Figure 4.19 (a) and (b)

respectively. The ports were labeled 1-12 according to the hour locations on an analog clock face with the northern most port designated 12.

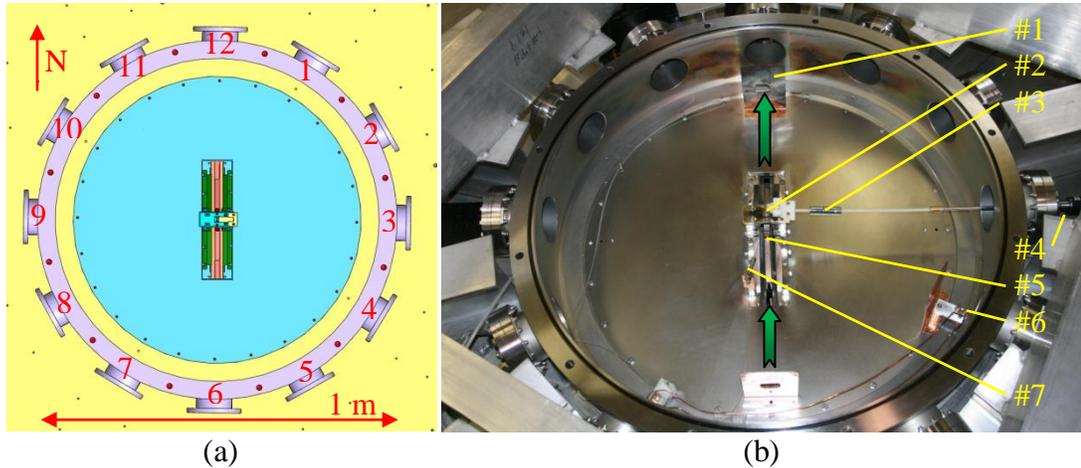


Figure 4.19 (a) A CAD rendering of the MAIZE experimental vacuum chamber with the support beams removed for clarity. The port numbers are labeled in red. (b) An image of the experimental chamber with the laser backlighter path shown in green. KEY: 1) Debris shields, 2) Foil load cartridge and tensioner, 3) Connection shaft for tensioning mechanism, 4) Rotating vacuum feed-through for tensioning mechanism, 5) MITL riser section, 6) B-dot current monitor, 7) Optical fiber.

The laser passed through the load region along the north-south axis, entering from port 6 and exiting from port 12. The windows on these ports were $\lambda/4$ fused silica from Kurt Lesker Company. Copper debris shields were installed to partially protect these windows from load plasma deposition, but these shields required holes for the laser backlighter beams to pass through, so the protection was for the unused portions of each window so that they could be periodically rotated to use an uncoated region before a complete cleaning was required.

A rotating vacuum feed-through was installed on port 3 and attached to a plastic threaded rod that actuated the foil tensioning mechanism. This tensioning mechanism allowed the foil to be restraightened after the chamber was evacuated. The tensioning mechanism was required to remove approximately 0.5 cm of compression at the load.

Because the threaded rod was plastic and only connected to grounded metal at the chamber wall, this current path was inductively and resistively isolated so that the drive current only passed through the comparatively negligible impedance along the MITL anode.

An ion gauge was installed on port 4 to monitor the high vacuum in the chamber and copper debris shield was installed in front of this gauge to protect it from the load plasma. This pressure reading was assumed to represent the vacuum pressure at the load and inside the MITL. Typical vacuum pressure before a shot was $\sim 50 \mu\text{Torr}$. Lower pressures were not required for the experiment and only served to increase the pumping time between shots.

A B-dot current monitor was installed in front of port 4 through a small hole in the anode plate to measure the drive current in the MITL. This device measured the voltage induced from the changing magnetic flux inside a wire loop. The drive current generated a significant time changing magnetic flux, so this device was well suited for measuring the derivative of the drive current as an induced voltage. This particular B-dot, constructed by M.R. Gomez and D.M. French, utilized a double-ended design where the induced voltage on each end of the same wire loop was measured on an oscilloscope as opposite polarity signals. These signals were then subtracted to remove any common mode noise that may have been generated on the cables and superimposed onto the B-dot signal. Since the B-dot directly measured the time changing magnetic flux rather than the time changing drive current, the induced voltage signal was only proportional to dI/dt , so an external calibration was required. The calibration factor for this B-dot was 3.97×10^{12} A/(V*s). The drive current itself was recovered by integrating the subtracted dI/dt signals

and then multiplying by the calibration factor. The same debris shield in front of the ion gauge also protected this B-dot. Three additional B-dot installation points in front of ports 1, 7, and 10 existed, but the port 4 B-dot was the only functional drive current diagnostic at the time of the MRT foil experiments. The B-dot signals travelled through 50 Ω semi-rigid coaxial cables to an SMA vacuum feed-through flange on port 7. Further details regarding the construction of these B-dot monitors are given in M.R. Gomez's PhD dissertation.

The foil load cartridge was installed on top of the rise section plates of the MITL in the center of the chamber. This positioning allowed the foil interaction region to be viewed from the north-south axis by the laser backlighter diagnostic. Four CeramOptec UV400/440 Tefzel-jacketed optical fibers were available inside the chamber. One fiber was positioned to view the load plasma to obtain preliminary spectroscopic information using a SpectraPro 750 spectrograph with a 150 groove/mm grating. Some preliminary spectroscopic information centered at 400 nm for shot 328 is provided in Appendix A.

4.2.3 Foil Load Cartridge Hardware

The Al foil used for the MRT experiments was $400 \pm 25\%$ nm thick. This foil was purchased from Goodfellow as 16x16 cm squares and cut into 1 cm wide by 3 cm long strips using sharpened and deburred razorblades. These fragile foil strips required a rigid cartridge to hold them in place across the LTD module AK gap and to form the desired double planar return current plate geometry. A series of CAD renderings of the foil load cartridge hardware are shown in Figure 4.20 and the detailed drawings of the hardware

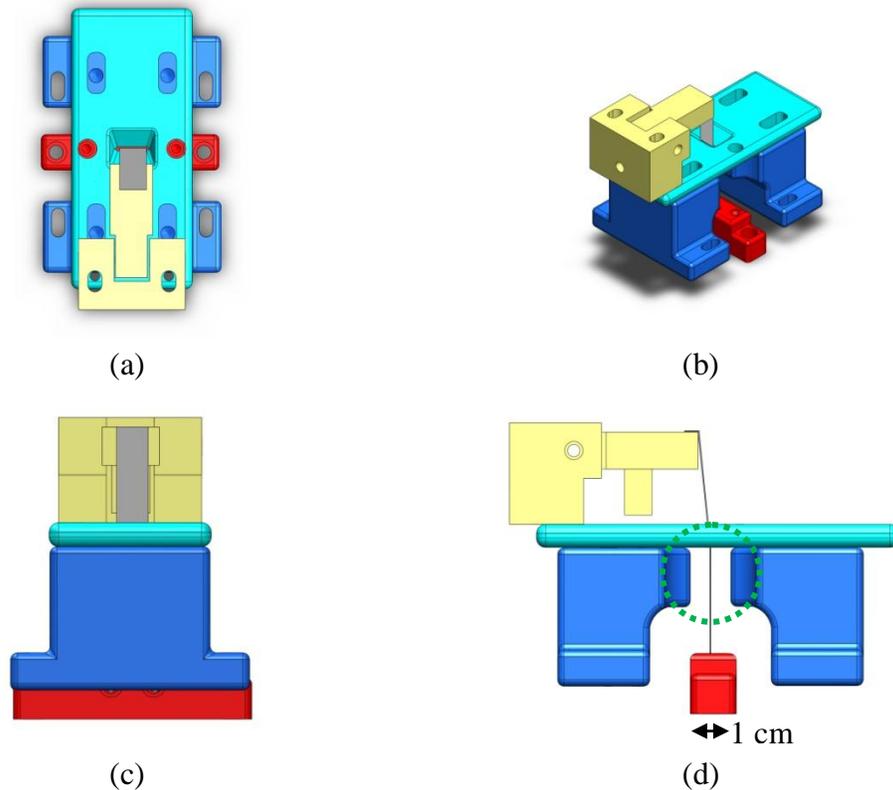


Figure 4.20 CAD renderings of the foil load cartridge hardware from the (a) top view, (b) isometric view, (c) front view, and (d) right side view. The anode return current plates are blue, the anode bridge is teal, the tensioner is yellow, and the cathode is red. The foil interaction region is circled in green.

components are provided in Appendix D. The cathode consisted of two brass pieces. The main piece secured to the cathode MITL riser plate and contained a small notch where the foil was attached. The second piece fit inside this notch and clamped the foil into the cathode using two counterbored screws. The foil extended vertically upward from the cathode across the AK gap to the anode bridge. This bridge was also made of brass and contained a narrow slot through which the foil physically passed while touching the edge. This electrically connected the foil to the anode. The foil was physically attached to an L-shaped electrically-insulating nylon lever arm using Al tape. This lever arm was hinged onto a nylon base piece that attached to the anode bridge. A plastic screw through the side of the base piece was attached to the rotating threaded rod from the rotating vacuum

feed-through on port 3. Images of the physical load cartridge hardware showing these components are provided in Figure 4.21. When the feed-through was rotated, the screw advanced and pushed on the shorter side of the hinged lever, which raised the longer side attached to the foil and allowed the foil to be tensioned from outside the vacuum chamber. The foil was tensioned until it was as straight as feasible when viewed through the laser window ports. The foil was free to slide through the anode bridge slot, but

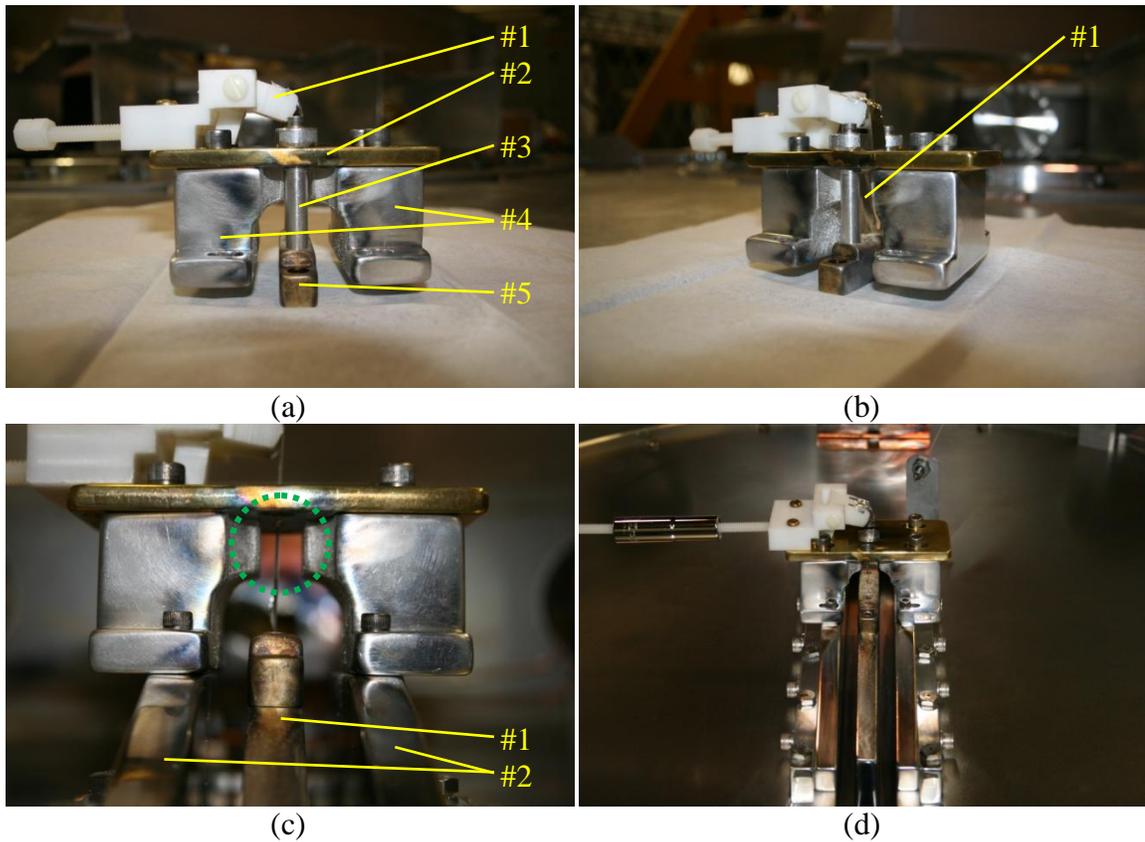


Figure 4.21 (a) A side view of the load cartridge before loading into the chamber with the support posts installed rigidly connecting the cathode to the anode bridge. The foil was obscured by the support post. **KEY:** 1) Tensioner with attached foil, 2) Anode bridge, 3) Support posts, 4) Anode current return plates, 5) Cathode. (b) An angled view of the load cartridge showing the foil inside the cartridge. Significant slack was left in the foil during transport to reduce the risk of tears. **KEY:** 1) Foil load. (c) A side view of the load cartridge installed in the experimental chamber with the support posts removed. The foil was only loosely tensioned for the photograph to reduce the risk of tears during the chamber evacuation process. The 1 cm wide interaction region was circled in green. **KEY:** 1) Cathode MITL riser plate, 2) Anode MITL riser plates, (d) The installed load cartridge showing the tensioning mechanism connected.

maintained electrical contact. The anode bridge connected to two Al anode pieces that formed the return current plates for the circuit. The through holes in the anode bridge for the connection bolts between the bridge and the current return plates were slotted. The anode pieces attached to the anode MITL riser plates using vented screws set through slotted channels in the anode pieces. These slotted connections between the anode bridge and the return plates, as well as between the return plates and the anode riser plates, allowed the return current plates to be moved relative to the foil, which was the method used to create either a small offset foil load or a large offset foil load. The cathode and anode bridge pieces were temporarily held together with support posts during the foil loading process and during transport to the experimental chamber. These posts, shown attached in Figure 4.21 (a, b), were removed after the hardware was bolted into both the cathode and anode MITL riser plates. Since the anode attachment points were slotted, the cartridge position was determined from the cathode mounting holes, which were not slotted, and the anode locations were fixed relative to the anode riser plates by the cathode location through the support posts.

The inductance for the foil load could not be calculated analytically because the geometry was too complex in the feed section below the interaction region. A static inductance was estimated using circuit model simulations, described in Chapter 5, to fit the simulated current to the experimental current measurements shown in Chapter 6. This inductance was approximately 4.5 nH. The actual load inductance was expected to contain a time-changing component, which was not possible in the circuit model, so the 4.5 nH represented a time-averaged inductance estimate. The exact static inductance for this region would have required a magnetostatic simulation to determine and then

integrate the complicated magnetic fields around the load to better calculate the static load inductance.

4.3 Laser Backlighter Diagnostic

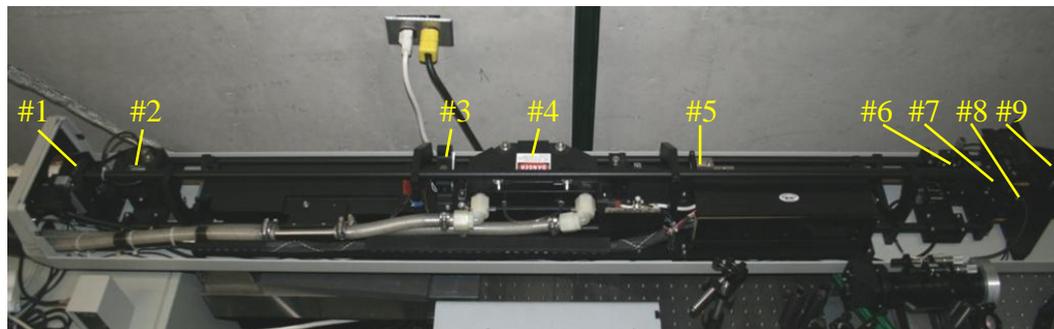
In order to image the plasma dynamics and to measure the MRT instability evolution, a very fast light source was required that could freeze the plasma motion occurring on a nanosecond timescale. This was accomplished by building a laser backlighter system that would act as the sub-nanosecond flash, combined with careful filtering to allow open shutter camera sensors to be used to collect the images. The laser system employed a neodymium-doped-yttrium-aluminum-garnet (Nd:YAG or YAG) crystalline rod as the gain medium pumped with a Xe flashlamp to produce 1064 nm laser light. This commonly used laser architecture allowed components to be easily acquired. The laser diagnostic was constructed in three locations: the front end, which contained the primary lasing cavity and pulse conditioning optics; the switchyard, which provided timing data, as well as split the beam along multiple paths to produce multiple image frames on a single foil shot using the same laser pulse; and the imaging section, which contained the focusing optics, filters, and camera sensors. The system was designed and constructed by J.C. Zier and D.M. French.

4.3.1 Front End Optics

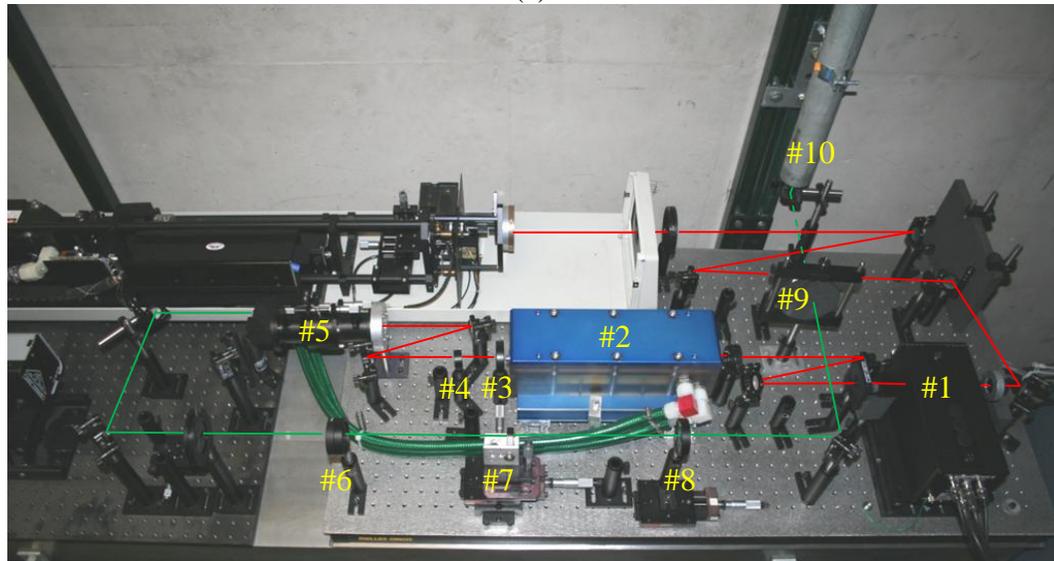
The primary front end beam path components consisted of a main lasing cavity, pulse extraction system, amplifier, second harmonic generation crystal, and spatial filter. An image of the main lasing cavity is shown in Figure 4.22 (a) and the beam path is shown in Figure 4.22 (b). These components combined to produce a sub-nanosecond

pulse of 532 nm light capable of freezing the motion of the foil plasma on an image to measure the evolution of the MRT instability.

The main lasing cavity was based on a Spectra Physics 3460 system. This cavity was mode-locked using a Spectra Physics 541 unit operating at 40.793 MHz. The cavity length was tuned using a micrometer-driven translation mechanism coupled to the high reflector mirror. The optical length between the high reflector and the output coupler



(a)



(b)

Figure 4.22 (a) An image of the Nd:YAG main lasing cavity. KEY: 1) High reflector and cavity length micrometer, 2) Q-switch optical head, 3) Electronic shutter, 4) Nd:YAG laser head enclosure, 5) Brewster window, 6) Mode-locker optical head, 7) Output coupler, 8) Internal photodiode, 9) Manual shutter. (b) An image of the front end beam path. KEY: 1) Pockels cell and crossed polarizers, 2) Amplifier optical head, 3) 50 cm focal length lens, 4) 1.0 neutral density in flip mount, 5) Frequency doubling optical head, 6) 25 cm focal length achromatic doublet lens, 7) 50 μm diameter aperture spatial filter, 8) 25 cm focal length collimating lens, 9) Removable beam dump, 10) Beam pipe to switchyard.

mirror was set to match this frequency, mode-locking the laser and producing a train of sub-nanosecond pulses every 12.3 ns. The specified FWHM pulse length for the mode-locked pulses was 70 ps. The cavity length had to be exactly tuned to prevent self-Q-switching. If the cavity was mistuned, the slight difference between the cavity resonant frequency and the mode-locking frequency would generate a beat frequency where larger amplitude pulses could build and introduce fluctuations in pulse amplitude. These fluctuations would have complicated the pulse extraction triggering and consequently would have increased the laser jitter with respect to the drive current. The current powering the flashlamp inside the laser head was set to 16.7 A. Larger currents in excess of 18 A were available and would have increased the energy in the pulses, but doing so was found to introduce a higher order spatial mode on the beam profile that would have resulted in additional losses in the spatial filter, and would have potentially introduced the nonuniformity onto the data images if the spatial filter did not fully remove the higher order mode. The lower lamp current was therefore chosen to improve the backlighter uniformity at the cost of pulse energy. The main cavity was also Q-switched using a Newport N33027-10-4 optical head driven by a Spectra Physics 3270 RF source. This system allowed RF power to be applied to a crystal inside the Q-switch optical head inside the lasing cavity that changed the beam alignment, reducing the cavity gain and allowing energy to store inside the YAG rod. When triggered, the RF signal was removed, which restored the cavity gain and allowed the pulses inside to extract the very large amount of energy stored in the gain medium. This resulted in a 300 ns long Q-switched envelope of laser pulses. The peak energy for these pulses was approximately

12 μJ . The laser system was therefore triggered by externally triggering this Q-switch unit to release a pulse envelope.

Outside the main lasing cavity the beam was turned using broadband dielectric mirrors coated for 750 nm – 1100 nm wavelengths. The reflectivity for 1064 nm at 45° incidence was 99.5%. The Q-switched pulse train contained many pulses, but a single pulse was needed for imaging the foil plasma, so a pulse extraction system was required. This system used a crossed pair of calcite polarizing beam splitters to nominally block the pulses. A Lasermetrics 5046ER optical head containing a Pockels cell was placed between the crossed polarizers. When triggered, a fast high voltage driver unit supplied an electric field across the Pockels cell. This rotated the polarization of the laser light between the polarizers in a manner proportional to the applied electric field. The voltage required to rotate the polarization 90° to pass the second polarizer was 8 kV. The high voltage was triggered using a voltage signal from the main lasing cavity's internal photodiode. The voltage trigger threshold and delay were adjusted so that when the Q-switch was triggered, the Pockels cell opened for a 10 ns window exactly when one of the large Q-switch pulses arrived at the pulse extraction system. This allowed a single pulse to be obtained for imaging the plasma whenever the laser was triggered. The Pockels cell driver was capable of operating up to 5 kHz. This was vital for laser alignment because a single pulse was not useful for aligning optics. For alignment mode the Q-switch was set to internally trigger at 200 Hz. Pulses were extracted from these envelopes in the same manner as for a single externally triggered Q-switched pulse, resulting in a beam suitable for aligning optics further down the beam path.

The extracted pulse from the Pockels cell did not contain enough energy to image the foil plasma due to the self emission levels from the foil itself, so an additional amplifier was employed. This second Nd:YAG rod and Xe flashlamp optical head were taken from a US Laser 408-1 system capable of producing a 750 W continuous wave laser beam. Unlike the laser head inside the main cavity that required continuous electrical energy powering the flashlamp, this amplifier only needed to be energized once to amplify the single pulse used to image the foil plasma. This was accomplished using a Korad flashlamp power supply. This supply charged a 420 μF capacitor to 2.1 kV. When triggered, this supply sent out a high voltage spike to break down the Xe flashlamp and then discharged the capacitor to energize the amplifier. The amplifier system is shown in Figure 4.23. This system was retrofitted with solid state relays to allow the *Laser Charge*, and *Laser Dump*, commands to be sent remotely from the DAQ. The voltage on the capacitor was monitored through a voltage divider across the capacitor electrodes. The *Laser Monitor* signal was sent through a 3.3 kHz lowpass RC filter and read by the DAQ system. The calibration for this signal is provided in Appendix E. The fire command required fast timing so this was handled by the Berkeley Nucleonics 575 delay generator. The amplifier discharge lasted for hundreds of microseconds, but required 250 μs to achieve full gain, so the laser amplifier was the first component triggered by the delay generator, and the Q-switch was triggered 250 μs later. Delay changes on the order of hundreds of nanoseconds did not affect the gain, so the Q-switch delay was varied without changing the amplifier delay. The amplifier gain for one of the extracted pulses was a factor of 183 with a single pass. The amplifier acted as a diverging lens so a

positive 50 cm focal length lens was placed after the amplifier to correct for the divergence.

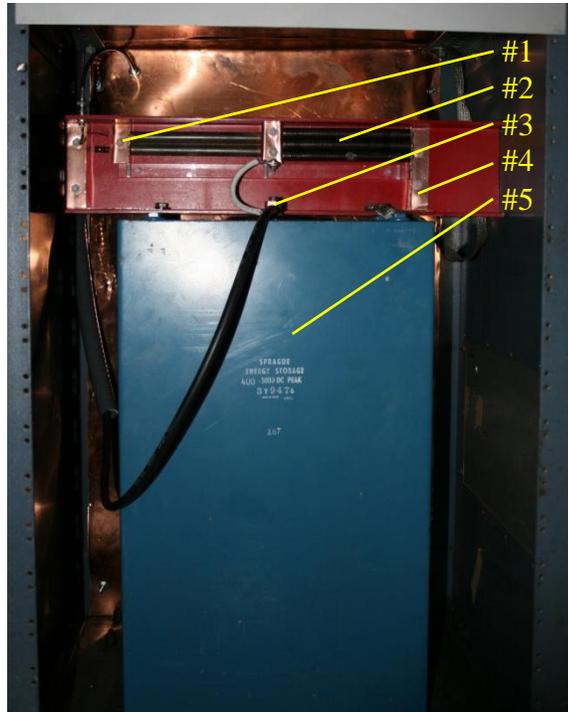


Figure 4.23 An image of the amplifier storage capacitor and monitoring system. **KEY:** 1) Voltage divider (~750:1) monitor, 2) $R = 550 \text{ k}\Omega$ bleedoff resistance. $RC = 230 \text{ s}$, 3) High voltage electrode attached to Korad supply, 4) Ground electrode, 5) $C = 420 \text{ }\mu\text{F}$ storage capacitor.

A 5 mm-square KTP crystal was placed after the amplifier section to double the laser frequency to 532 nm green light suitable for collection by the camera sensors and improving the penetration capability of the beam through the plasma. A dichroic mirror separated the green second harmonic light from the infrared fundamental. For alignment operations the laser was sent directly into the frequency doubling crystal. For amplified operation, the 0.8 mm radius beam exceeded the 250 MW/cm^2 damage threshold for the KTP crystal, so a 1.0 neutral density filter was needed after the amplifier to reduce the laser intensity to $\sim 225 \text{ MW/cm}^2$. This added attenuation could have been reduced by

further expanding the beam radius on the crystal, but the exact amplifier performance was not known when the laser was being constructed, and it was unclear how much the intensity could have been reduced at the crystal while still maintaining second harmonic generation in alignment mode.

After the frequency doubling section the laser was manipulated with broadband dielectric mirrors coated for 400-700 nm. The laser passed through a spatial filter consisting of two 25 cm focal length lenses spaced approximately 50 cm apart to send the laser through a focus and then recollimate the beam. A 50 μm diameter high damage threshold pinhole was placed at the focus between these lenses. This aperture acted as a lowpass filter for the spatial modes on the wave front, leaving a clean Gaussian profile. This extracted, amplified, frequency doubled, and cleaned laser pulse was then directed by mirrors through a series of tubes onto the switchyard optical table near the LTD module.

4.3.2 Switchyard Optics

The switchyard section, shown in Figure 4.24, was responsible for the final optical manipulations of the laser pulse before sending it through the experiment. The laser was received from the beam pipes, sent through an alignment iris, and then expanded by a factor of 5 using 10 cm and 50 cm focal length lenses spaced 60 cm apart, resulting in a roughly 2 cm diameter beam. A 1% beam sampler was used to send a small amount of laser energy onto a photodiode for timing information. The main laser pulse was then divided using a 50/50 beamsplitter. The reflected light from this optic was immediately sent to a pair of targeting mirrors and then to the experiment. The second

beam was bounced off two additional mirrors on the optical table in order to add a 9 ns delay between this beam and the first. After traversing the additional path length, the

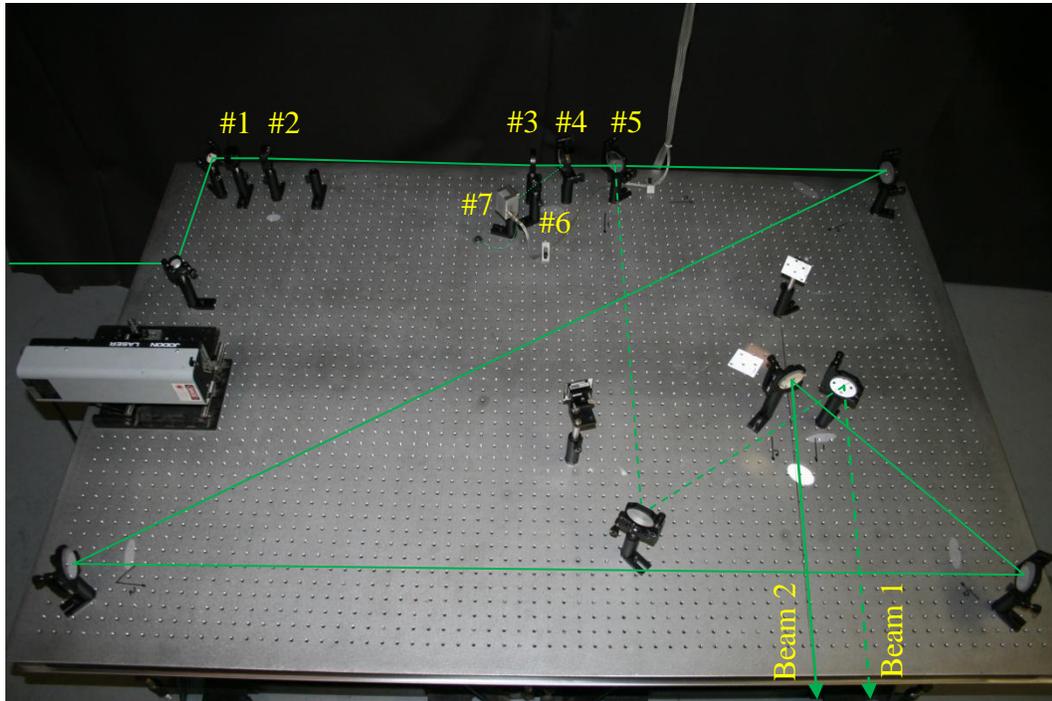


Figure 4.24 An image of the switchyard optical table and beam path. **KEY:** 1) Alignment iris, 2) 10 cm focal length lens, 3) 50 cm focal length lens, 4) 1% beam sampler, 5) 50/50 beamsplitter, 6) 5 cm focal length lens, 7) Timing photodiode.

second beam was sent to another pair of targeting mirrors and then to the experiment. The final targeting mirrors for each beam path were symmetric about the optical axis through the load, and the several meter standoff distance between the optical table and the chamber, as well as the choice of 1 m focal length imaging lenses in the imaging optics section, allowed the offset angle for each beam from the optical axis to be a negligible 0.86° . This ensured that both laser beams were able to travel straight through the load. The diagnostic path through the load region is shown in Figure 4.25. The total delay times between when the oscilloscope measurement from the photodiode occurred and

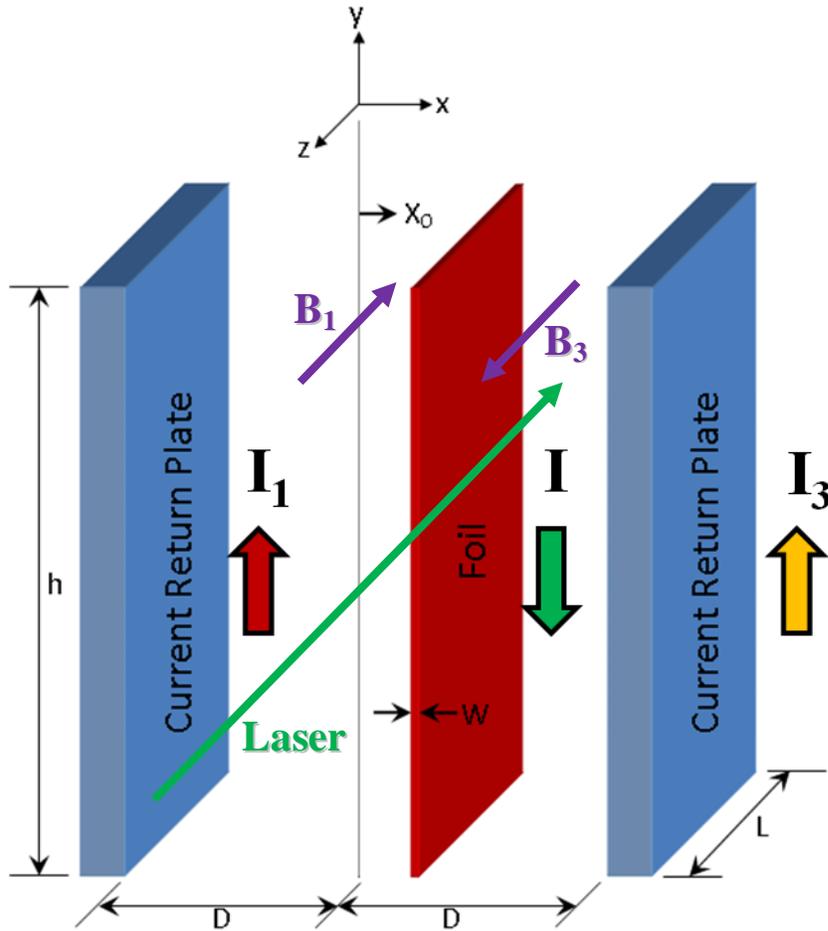


Figure 4.25 The geometry for the planar foil load region showing the magnetic fields and laser backlighter path. This optical path measured the y-direction MRT wavelength modes.

when each laser passed through the load, including electrical cable delay and laser propagation, were 84.5 ns and 75.5 ns for beam 1 and beam 2 respectively.

4.3.3 Imaging Optics

The imaging optics section was responsible for isolating the laser light from both beam paths and imaging the foil plasma onto the camera sensors. The background emission from the plasma was a considerable challenge to overcome in order to achieve an acceptable signal to background contrast level on the images. This was achieved by placing all of the imaging section optics inside a light-tight enclosure and heavily

filtering the small openings for the laser beams. An image of the beam path is shown in Figure 4.26. Each beam passed through a 1 nm FWHM line filter for the 532 nm laser wavelength. This blocked out the vast majority of the broadband background emission. A 0.4 neutral density filter was also used at the enclosure entrance to further reduce the light levels. The 1 m focal length, 5.08 cm diameter, achromatic doublet imaging lenses were placed 1.85 m from the load. Achromatic doublet lenses were chosen even though the backlighter laser was monochromatic because the doublets also reduced spherical aberration, resulting in tighter focal spots and better images than achievable with singlet lenses. A pair of receiving mirrors directed each beam onto its respective camera sensor. In front of each camera an additional 0.1 neutral density filter and 10 nm FWHM 532 nm line filter was used to further improve the image contrast level and to avoid saturating the sensor.

The cameras themselves were Canon Digital Rebel XSI single lens reflex models with 22.2 mm by 14.8 mm CMOS sensors. The cameras were placed 215.9 cm from the lens, resulting in a -1.16 magnification factor. The load region was directly imaged onto the camera sensors so each image was a focused shadowgraph. Each pixel represented $4.48 \pm 0.07 \mu\text{m}$ at the load. The uncertainty in the pixel scale represented the uncertainty in the magnification. The Rayleigh limited resolving power of the imaging system was

$$\Delta l = 1.220 \frac{f\lambda}{D}, \quad 4.12$$

where Δl was the diffraction limited resolution, f was the imaging lens focal length, λ was the laser wavelength, and D was the width of the laser escaping the interaction region.⁵⁸ For $f = 1\text{m}$, $\lambda = 532\text{ nm}$, and $D = 1\text{ cm}$ the resolution was $65 \mu\text{m}$. This was nearly 15

pixels on the imaging sensor, which indicated that the sensor resolution was more than significant for resolving the smallest features that the optical system was capable of

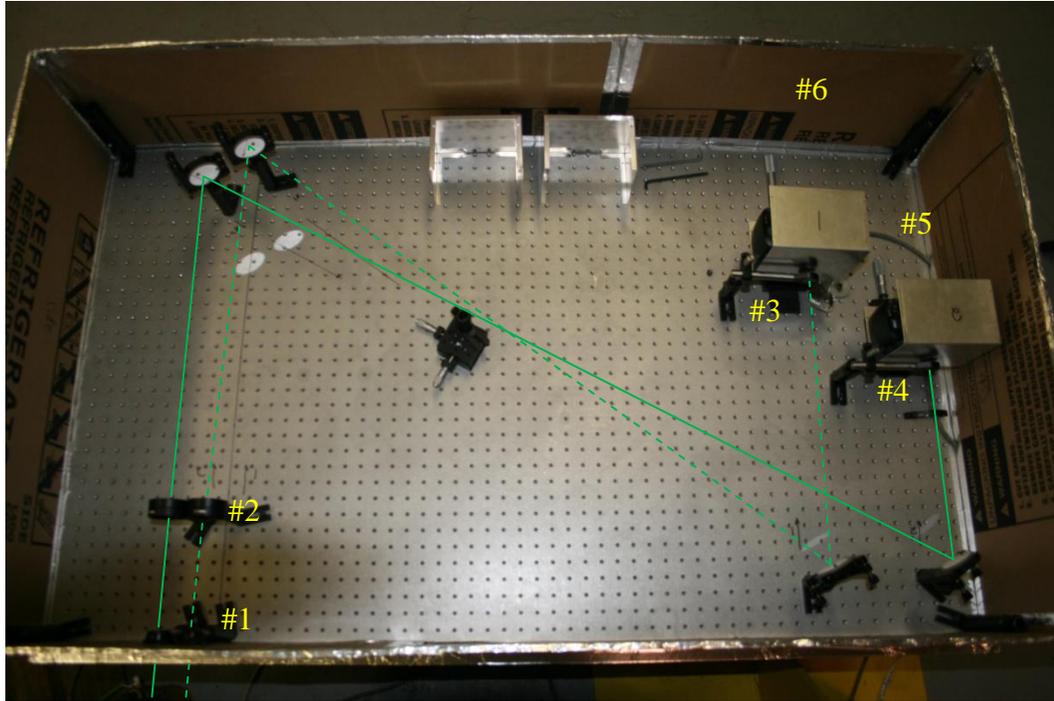


Figure 4.26 The imaging optics section for the laser backlighter diagnostic. **KEY:** 1) 532 nm line filter (1 nm FWHM) and 0.4 neutral density filter for each beam path, 2) 1 m focal length achromatic doublet imaging lenses for each beam path, 3) Camera 1 assembly complete with 0.1 neutral density filter, 532 nm line filter (10 nm FWHM), Canon XSI digital camera, and Faraday shield enclosure, 4) Camera 2 assembly (see #3), 5) Remote trigger cables, 6) Light tight enclosure with lid removed.

resolving. Since the lens focal length was 100 times longer than the width of the interaction region, the uncertainties in the measurements presented in Chapter 6 were dominated by the Rayleigh limited resolution. The uncertainty in the pixel scale became comparable to the diffraction limit only for measurements approaching 1000 pixels long, which was almost half the width of the interaction region. The cameras were operated in manual mode using a 100 ISO and an open shutter. The camera shutters were opened before the shot triggered using the *Camera Trig* signal from the DAQ. This signal closed a shorting relay, which shorted external trigger pins 1 and 3 together on the cameras to

hold the shutter open as long as the *Camera Trig* signal was active. The cameras were shielded from RF noise by grounded metal enclosures acting as Faraday shields.

CHAPTER 5

EXPERIMENTAL PERFORMANCE PREDICTIONS USING A CIRCUIT MODEL DRIVE PULSE

A method for predicting the performance of the MAIZE LTD module was needed for transmission line and load hardware design. The solution for fulfilling this need was a circuit simulation model that used a variant of SPICE, LTSPICE IV, developed by Linear Technology Corporation.⁵⁶ This particular SPICE implementation was chosen because it included support for nonlinear ferromagnetic inductors. A complete circuit model of the LTD module was constructed using a block-based structure of subcircuits to form the complete model. The LTD architecture lent itself well to this construction approach because the brick subunits, described in Section 4.1.1, were all identical and the LTD module contained significant symmetry. The block architecture allowed any static transmission line and load circuit to be attached to the LTD module leads to test the performance. This model was verified against known matched load experimental measurements, and then was configured with the MAIZE transmission line. The dynamic foil load characteristics were not known, so the load resistance and inductance values were empirically fit to the experimental data to match the experimental current peak and risetime. The experimental data were collected to calibrate the model at the lowest charge voltage available at which the LTD module would still perform properly. The model was then used to gauge the effect of higher charge voltages on the module capacitors and AK

gap insulator. The simulated drive current was also used to make foil load performance estimates using the expansionless slab model (ESM) described in Section 3.2.

5.1 SPICE Circuit Model

The circuit model was constructed by first building the circuit for a brick using the capacitances, inductances, and resistances detailed in Section 4.1.1. Circuit elements were employed symmetrically to maintain the top-bottom symmetry of the brick components about the trigger plane, resulting in a semi-physical representation of the actual bricks. This symmetry also simplified debugging the model because any deviations from symmetry indicated an error in the circuit. The bricks were connected in parallel banks of 10 to represent a quadrant of the LTD module. This was done to mimic the quadrant triggering pattern described in Section 4.1.5. The actual LTD module quadrants were connected through the same azimuthal trigger wire in the midplane insulator, but constructing the circuit model in this manner maintained the flexibility to independently trigger each quadrant of the machine. This option would prove very useful if the physical LTD triggering wire was split to allow independent quadrant triggering of the module for experiments that required a shaped pulse. The four quadrant blocks were connected in parallel for the assembled LTD module. All individual component diagrams are provided in Appendix C along with their netlist commands, and the highest level of the assembled LTSPICE model is shown in Figure 5.1. LTSPICE allowed circuit elements to contain additional basic circuit elements in series and in parallel with the main element, such as capacitor series resistance, instead of requiring these to be added separately, which improved the code performance. Unfortunately, this feature also made these sub parameters not immediately visible on images of the circuit diagrams, such as those

documented in Appendix C. The complete circuit element parameters given in Section 4.1.1 must be consulted when referencing these diagram images. This is a nonissue when using the actual simulation code, or when referencing the provided netlists. The highest assembly level contained the control options for the simulation and connected the power supplies, ferromagnetic cores, and triggers to the LTD module block. Transmission line and load elements were attached to the anode and cathode leads. The positive and negative power supply voltages were controlled in this assembly and could be changed by adjusting only two numbers instead of requiring each capacitor to be initialized with a different initial charge.

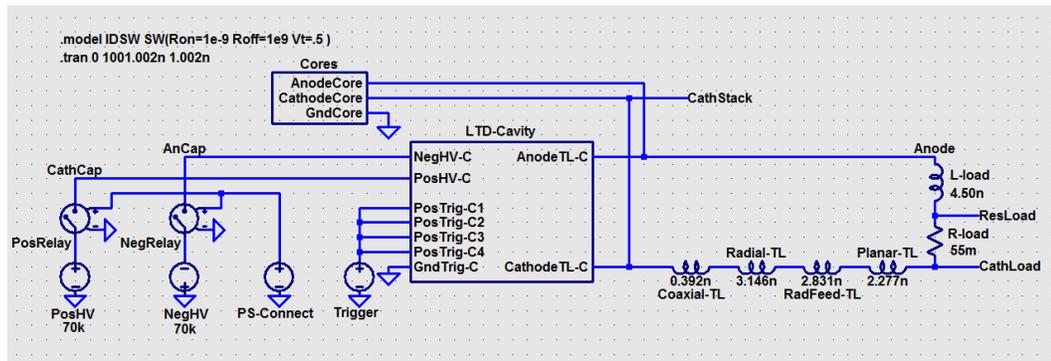


Figure 5.1 Control level for the assembled LTSPICE model of the LTD module.

The ferromagnetic cores were modeled as nonlinear inductors with a 1.5Ω effective resistance in parallel with the core inductance.⁶ The inductors were configured with the hysteresis parameters described in Section 4.1.2 and were modeled by LTSPICE according to Equations 4.2 and 4.3. The cores themselves were split about the circuit midplane to mimic the top and bottom physical cores of the LTD module and to preserve symmetry.

One limitation on SPICE simulations is that there is difficulty maintaining high temporal resolution of fast transients, such as a 100 ns risetime LTD pulse, occurring

within a longer simulation, such as a 40 μs risetime premagnetization pulse, while maintaining a reasonable timestep. This computational limitation prevented the capacitors from being charged and the ferromagnetic cores from being premagnetized using the physical timescales for these processes, which was not necessarily a desirable feature for charging, but was a desirable feature for premagnetization. This computational limitation was overcome using a computational workaround that set all the initial conditions in less than a nanosecond. The power supplies used to charge the machine were perfect voltage sources, so they could supply arbitrary current to charge the LTD capacitors during the nanosecond initialization time. The premagnetization circuit was constructed using the same principle. According to Equation 4.4, the given magnetic cross-section only needed to experience the same volt-second integral, so the voltage was increased and the application time decreased from tens of microseconds to the simulation initialization timescale. The 64 cm^2 total core area needed to be premagnetized from zero magnetization to the -2 T saturation point. This was accomplished by applying a 25.6 MV potential for 0.5 ns. The cores needed to relax 0.8 T to reach the remanent embedded magnetization for the physical cores before a shot, so an opposite polarity 10.2 MV potential was applied for 0.5 ns. After the 1 ns initialization time, all the charging power supplies were switched out of the circuit, leaving the capacitors and ferromagnetic cores with the proper initial conditions, and allowing the 1000 ns long simulation to proceed using reasonable time step resolution. No data were recorded during the initialization portion of the simulation, so only values resulting from the physically accurate circuit elements were retained.

5.2 Matched Load Performance

An approximately matched load was evaluated using the same 0.1Ω resistance in series with a 1.05 nH inductance as was used to test the physical LTD module at the Institute of High Current Electronics.⁵⁵ The voltage and current through the load, the voltage reversal across the capacitors, and the AK gap insulator “stack” voltage were recorded. Charge voltages of $\pm 70 \text{ kV}$ and $\pm 80 \text{ kV}$ are shown in Figure 5.2, and charge voltages of $\pm 90 \text{ kV}$ and $\pm 100 \text{ kV}$ are shown in Figure 5.3. The load voltages were

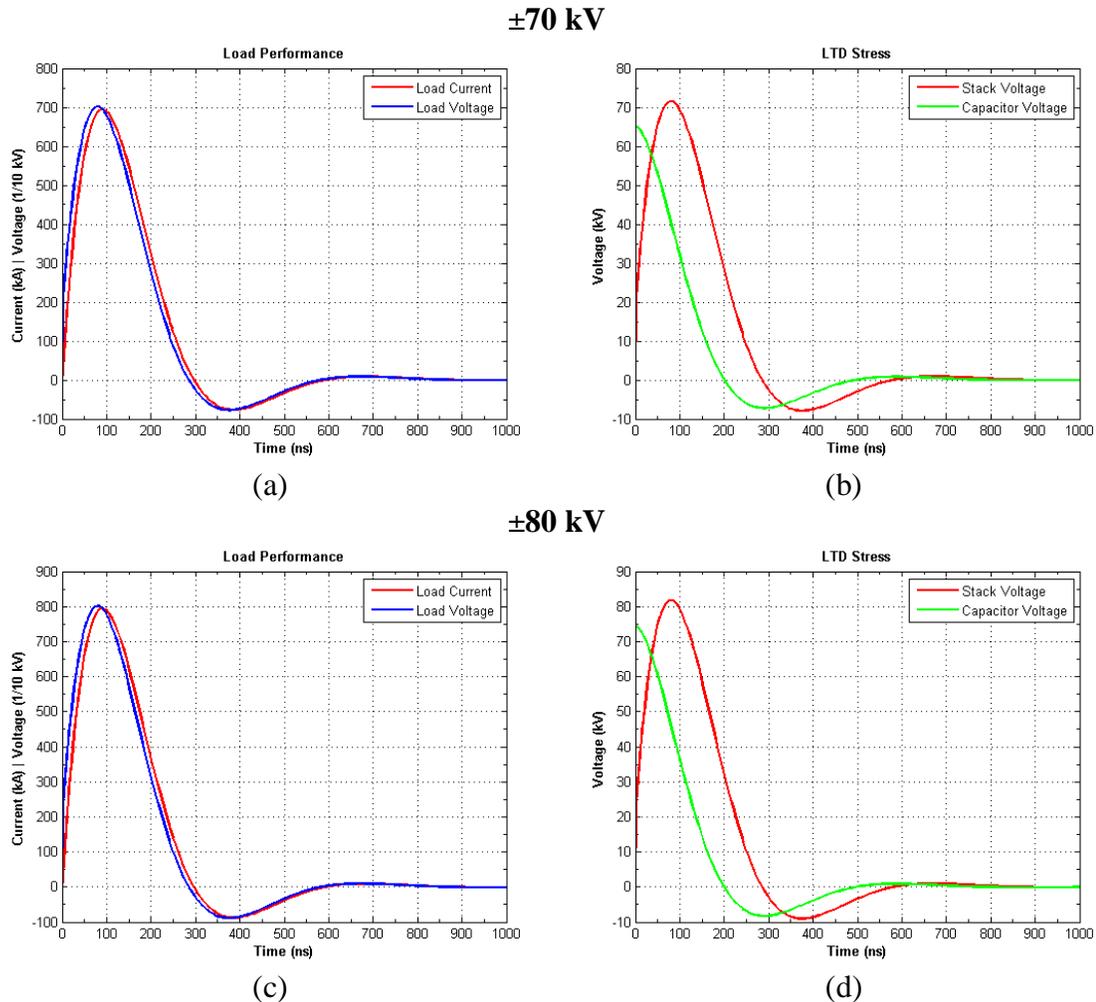


Figure 5.2 (a) The load current and voltage (x10) performance for $\pm 70 \text{ kV}$ charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal for $\pm 70 \text{ kV}$. (c) The load current and voltage (x10) performance for $\pm 80 \text{ kV}$ charge voltage. (d) The AK gap insulator stack voltage and capacitor voltage reversal for $\pm 80 \text{ kV}$.

multiplied by 10 to put them on the same scale as the drive current. The current and voltage traces were almost perfectly in phase due to the small 1.05 nH inductance, and the undershoot was approximately 10% of peak voltage. The small undershoot limited the voltage reversal on the capacitors, which were only rated to 10% reversal at the maximum 100 kV charge. The low voltage reversal rating was an important consideration when diagnosing the electrical stress on the LTD module for larger charge voltages. Decreasing the charge voltage increased the allowed voltage reversal on the capacitors:

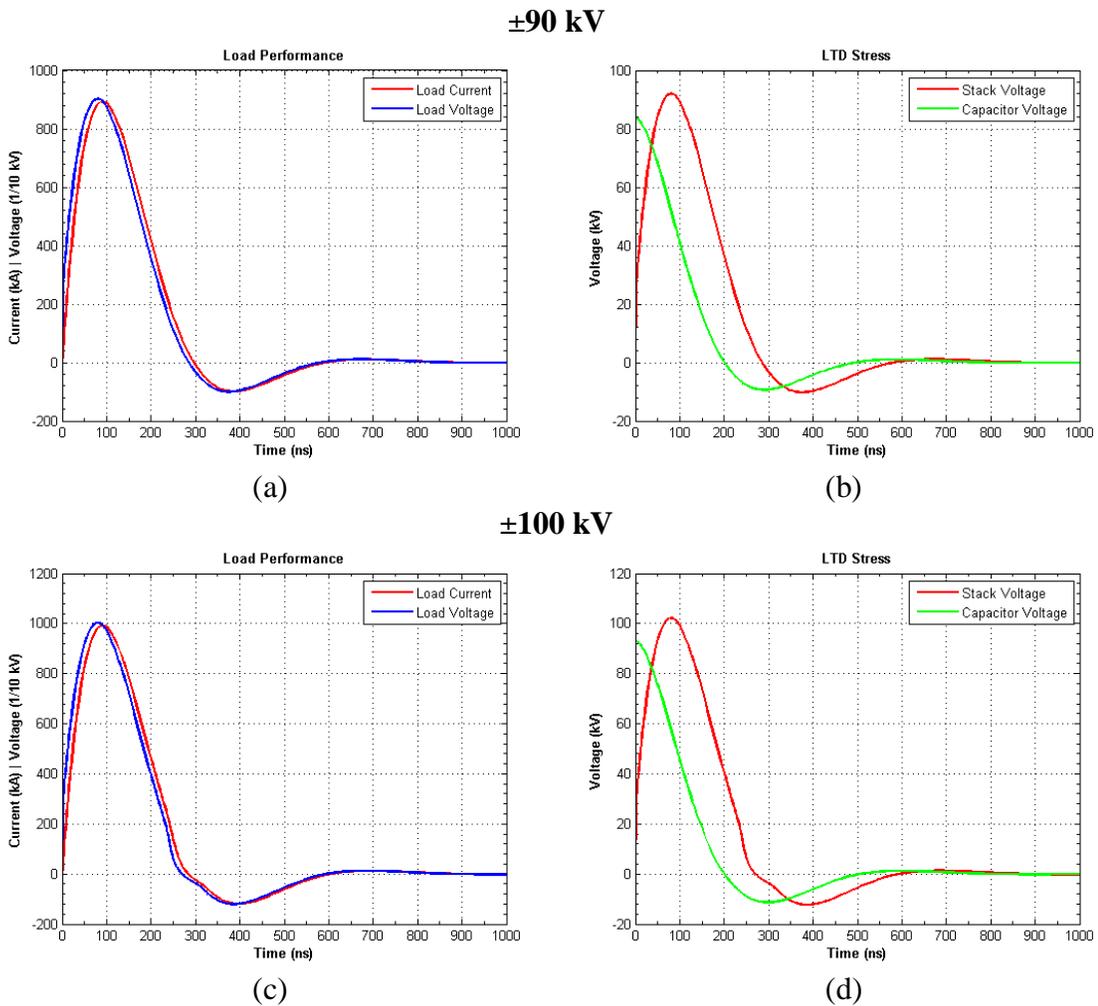


Figure 5.3 (a) The load current and voltage (x10) performance for ±90 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal for ±90 kV. (c) The load current and voltage (x10) performance for ±100 kV charge voltage. (d) The AK gap insulator stack voltage and capacitor voltage reversal for ±100 kV.

10% at 100 kV, 25% at 90 kV, 45% at 80 kV, and 65% at 70 kV. The voltage across the AK gap insulator only reached the 100 kV design specification at the full ± 100 kV charge voltage. This was not surprising because the insulator was designed to operate at ± 100 kV charge voltage into a matched load. The peak current for the ± 100 kV case reached 990 kA with a 90 ns risetime, and the voltage across the load reached 100 kV as expected. These results exactly matched the experimental and SPICE results obtained at the Institute of High Current Electronics.^{6,42,55} which served as a strong validation of the new LTSPICE model.

The cores were designed to not saturate during the main drive pulse for a matched load. The only detectable saturation was at the very end of the ± 100 kV pulse around 250 ns where the voltage abruptly dropped.

5.3 Foil Load Performance with the Expansionless Slab Model

The LTSPICE model was reconfigured with the 8.65 nH inductance estimate for the MAIZE transmission line. The foil load resistance and inductance were empirically fit to ± 70 kV current traces such as those shown in Figure 6.9. The static load values were 4.5 nH and 55 m Ω . It must be stressed that these empirical values were in series with the transmission line, so any deviations in those estimates from the physical values would alter the static load values. Additionally, the load values represented the time-averaged values and did not include any dynamic load effects. The purpose of the static load values was to aid in evaluating the overall electrical stress on the LTD module components rather than to diagnose the load physics. The 620 kA and 170 ns risetime for the ± 70 kV case agreed with the experimental current measurements, which allowed the simulated stresses on the capacitors and insulator to be used to evaluate higher charge voltage risk.

The simulated drive currents were used to examine the dynamic load predictions from the expansionless slab model described in Section 3.2. The foil offset was 2 mm for all charge voltages and the foil dimensions were 400 nm thick and 1 cm wide.

5.3.1 Performance with ± 70 kV Charge Voltage

The inductance for the foil load and transmission line was 13 times larger than for the matched load case. This had a profound impact on the LTD module performance as shown in Figure 5.4. The peak current was reduced from 700 kA to 620 kA, and the risetime lengthened from 90 ns to 170 ns. Additionally, the load voltage only reached 42

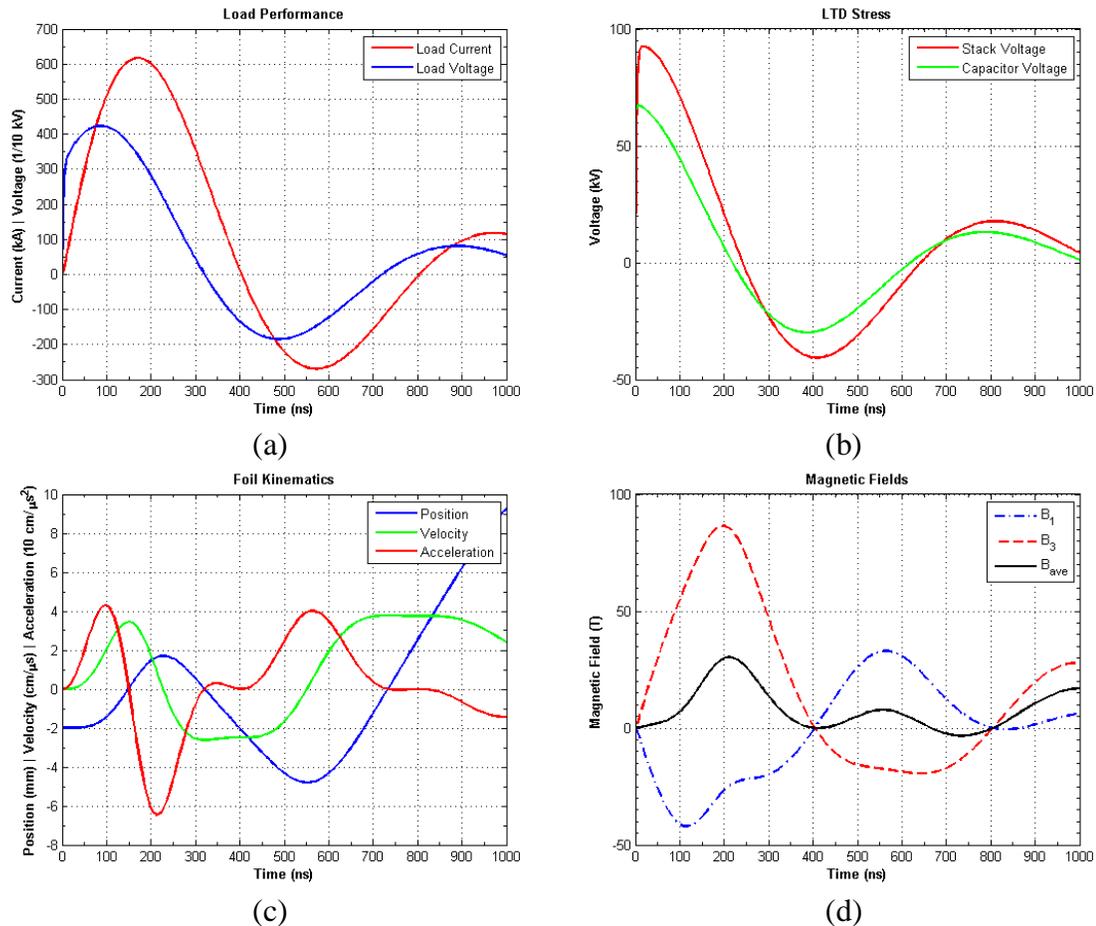


Figure 5.4 (a) The foil load current and voltage (x10) performance for ± 70 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.

kV and was no longer in phase with the current. This reduced the power coupling to the load, introduced a 50% undershoot, and introduced significant ringing after the main drive pulse. The voltage reversal on the capacitors was 45%, which was within the design specification for 70 kV operation. The insulator voltage reached 92 kV, which was approaching the 100 kV design specification. The magnetic field reached 86 T.

5.3.2 Performance with ± 80 kV Charge Voltage

Increasing the charge voltage to ± 80 kV introduced added stress on the LTD module components as shown in Figure 5.5. The capacitor voltage reversal remained at

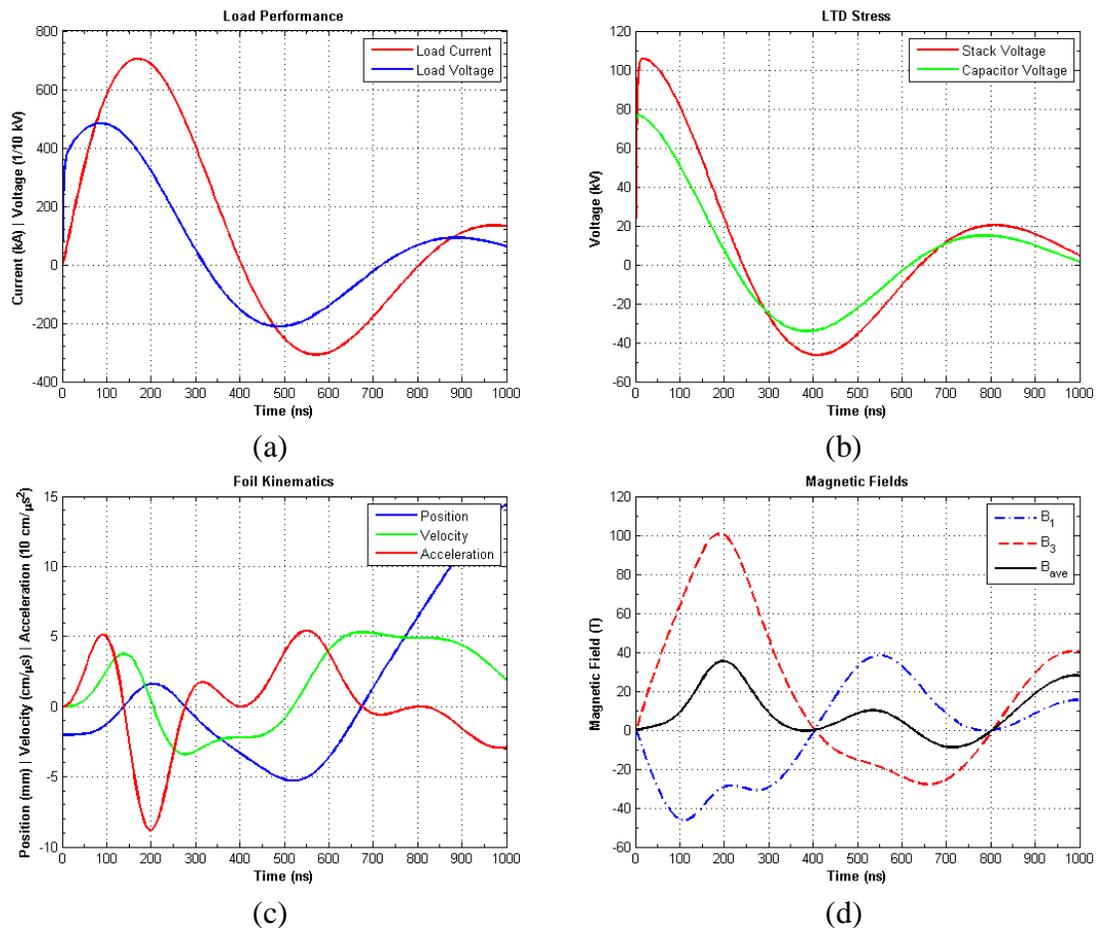


Figure 5.5 (a) The foil load current and voltage (x10) performance for ± 80 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.

45 %, but this was exactly at the reversal rating for an 80 kV charge. The insulator voltage increased to 106 kV, which was above, but still close to the 100 kV rating. The load current increased to 705 kA and the voltage increased to 48 kV. The cores showed no sign of saturation, and the magnetic field increased to 100 T. These results indicated that the LTD module could have been operated at ± 80 kV, but doing so would have pushed the components harder than was necessary to perform the MRT experiment, so the charge voltage was kept at ± 70 kV.

5.3.3 Performance with ± 90 kV Charge Voltage

Increasing the charge voltage to ± 90 kV resulted in significant stress on the LTD module and saturated the cores at 200 ns, as shown in Figure 5.6. The current reached 800 kA, but the voltage on the insulator peaked at 120 kV. This was 20% above the 100 kV rating. Additionally, the 46% voltage reversal was nearly double the 25% voltage reversal rating at 90 kV. The load voltage peaked at 55 kV and the magnetic field peaked at 115 T. Although the added current provided a larger acceleration for the foil, the greater electrical stresses on the LTD module components ruled out physical LTD shots at ± 90 kV with a foil load. The inductance of the transmission line, and potentially the load, would need to be reduced in order to take advantage of the larger drive currents accessible using larger charge voltages.

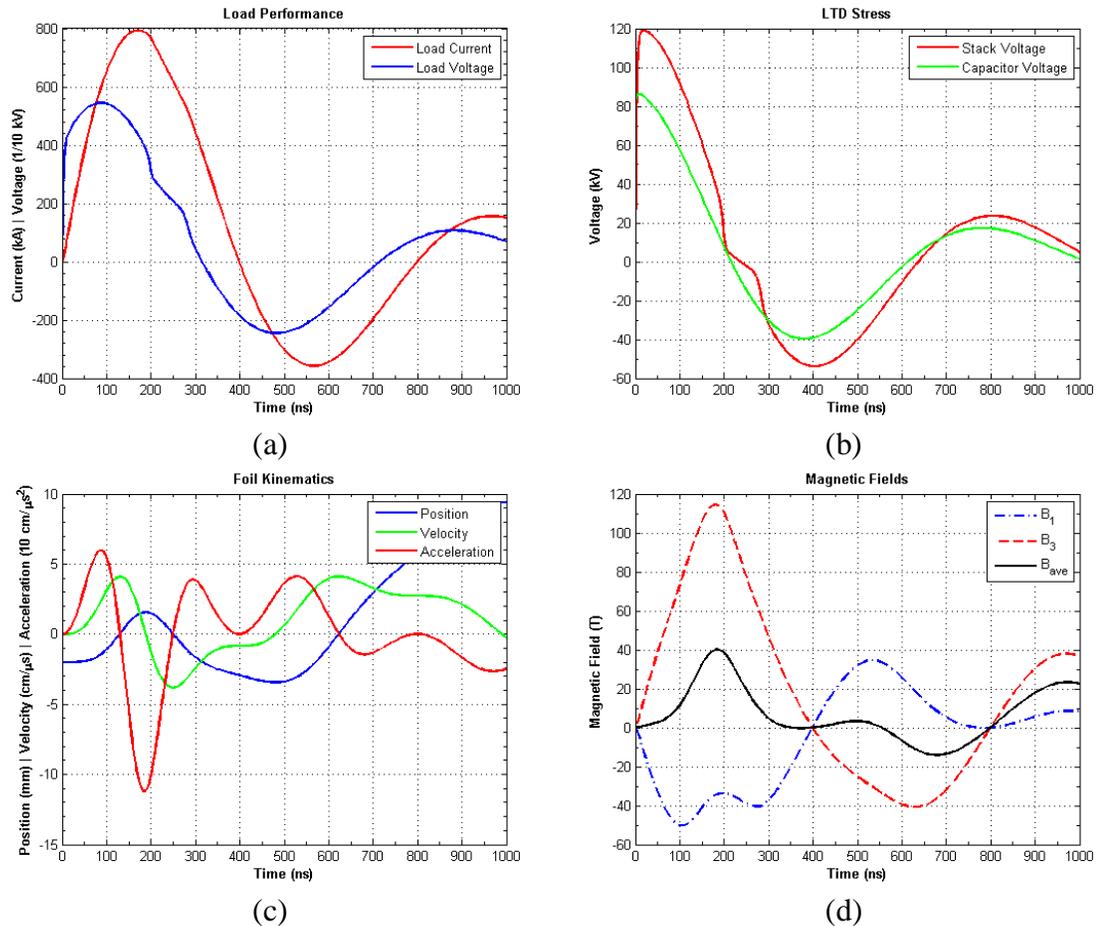


Figure 5.6 (a) The foil load current and voltage (x10) performance for ± 90 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.

5.3.4 Performance with ± 100 kV Charge Voltage

As expected from the ± 90 kV case, increasing the charge voltage to the LTD module's matched load capacity resulted in dangerous electrical stresses on the components. Although the load current reached 873 kA, the voltage reached 61 kV, and the peak magnetic field was 125 T; the voltage reversal on the capacitors was 51%. This was over 5 times the rated voltage reversal for a 100 kV charge and would have resulted in a significant lifetime reduction for the capacitors. The insulator voltage peaked at 132 kV, over 30% above the designed voltage level. Additionally, the now considerable voltage reversal across the insulator reached 66 kV, which would have generated an

electric field in the opposite direction from the intended polarity for the dielectric angle. This reversal would have greatly increased the risk of insulator flashover even if the insulator held off the larger initial field.

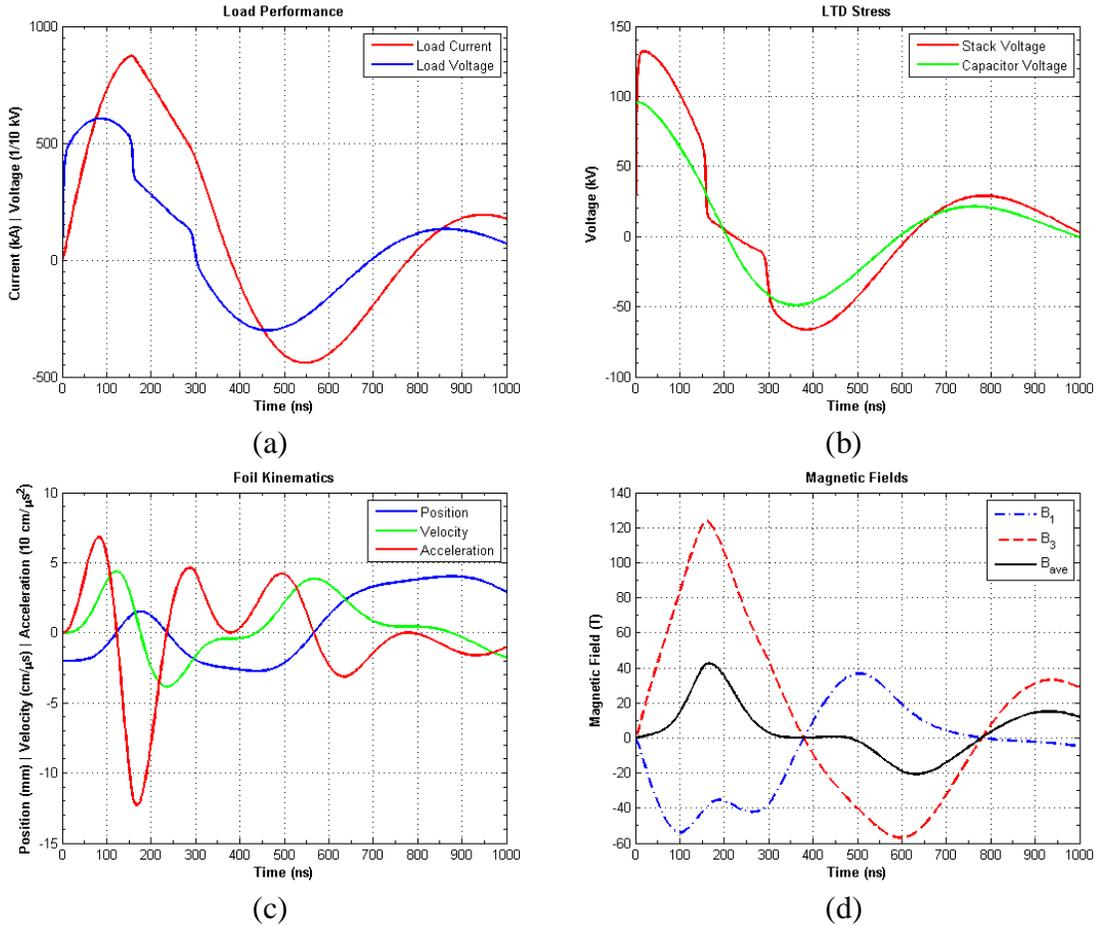


Figure 5.7 (a) The foil load current and voltage ($\times 10$) performance for ± 100 kV charge voltage. (b) The AK gap insulator stack voltage and capacitor voltage reversal. (c) The predicted foil kinematics from the expansionless slab model. (d) The predicted magnetic fields on either side of the foil.

5.4 Foil Load MRT Growth Predictions from the Expansionless Slab Model

The ESM foil accelerations for the ± 70 kV, ± 80 kV, ± 90 kV, and ± 100 kV cases were used to calculate the resulting instantaneous MRT growth rates, γ , and the integrated number of e-foldings, Γ . The Rayleigh-Taylor growth scaling, given by Equation 3.118, along with the generalized 3-region scaling, given by Equation 3.144

assuming $k_y = k_z$ (y is the direction of the current on the foil, and z is the direction of the magnetic field), were calculated and shown in Figure 5.8. The y -direction wavelength was set at 1 mm as measured in Section 6.3.3. The absolute value for the Rayleigh-Taylor scaling was used instead of just the universally correct imaginary component to signify 100% instability feed-through due to the extremely thin 400 nm foils used. For growth on the initially unstable interface, the real component happened to correspond to when the foil was accelerating in the opposite direction. This interface was stable during this motion, and the opposite interface was unstable, but 100% feed-through transferred this

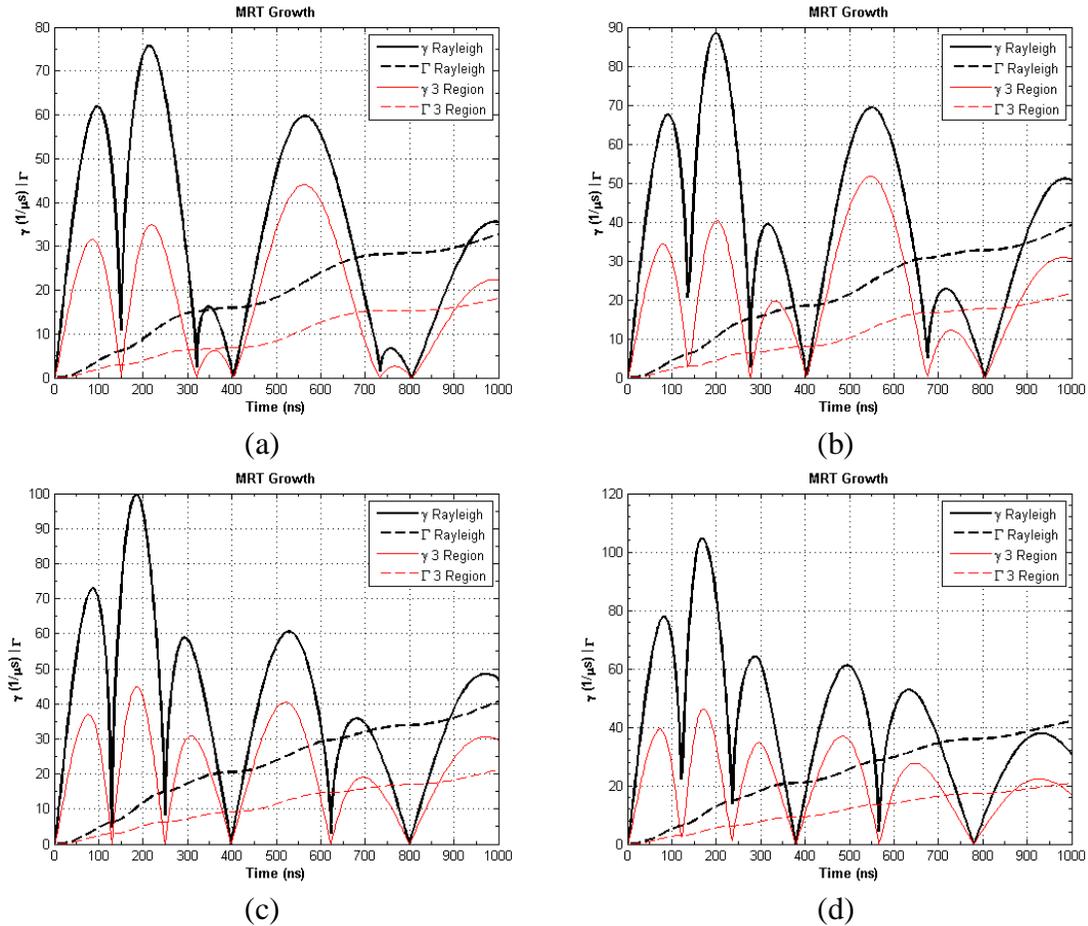


Figure 5.8 The predicted instantaneous MRT growth rates and integrated e-foldings for (a) ± 70 kV, (b) ± 80 kV, (c) ± 90 kV, and (d) ± 100 kV charge voltage. The Rayleigh-Taylor and generalized 3-region scalings are shown to illustrate the damping effect from the magnetic field.

instability growth to both interfaces regardless of the acceleration direction, which was mathematically equivalent to taking the absolute value. The z-direction wavelength was assumed equal to the y-direction wavelength for the sole purpose of illustrating the instability reduction from bending the magnetic field lines along the z-direction. The z-direction wavelength could not be measured experimentally using the hardware described in Section 4.2.

The predicted e-folding times from the ESM accelerations were significantly shorter than the measured e-folding times due to plasma expansion in the experiments that severely limited the validity of the ESM. The ± 70 kV ESM e-folding time was 13.3 ns, compared to the 100 ns measurements from the experiment. The expansion, described in detail in Section 6.5.3, caused the foil to not accelerate as a single slab, so the ESM accelerations that were used to calculate the MRT growth rates in Figure 5.8 were too large and resulted in much shorter e-folding times than the measured values.

CHAPTER 6

MAGNETO-RAYLEIGH-TAYLOR EXPERIMENTAL RESULTS

The pulsed power, load cartridge, and two frame laser backlighter diagnostic experimental configuration described in Chapter 4 was used to measure the magneto-Rayleigh-Taylor (MRT) instability growth rate on a 400 nm thick Al foil in a planar geometry. Two initial foil displacements from the midplane between the anodes were tested. One case utilized roughly a 1 mm offset to generate an acceleration toward the midplane, referred to as the large offset case, and is described in Section 6.1. The second case utilized a foil displacement that was as close to the midplane as was achievable given physical hardware constraints. This case, referred to as the small offset case, is described in Section 6.2. Details regarding the quantitative image analysis are provided in Section 6.3. Sample images taken at the same time from each case are used to illustrate the analysis used for the entirety of both offset data sets, which are presented in summary plots for each case. The MRT instability growth rate measurements, comparison of these measurements to the expansionless slab model (ESM) predictions, and an analysis of the results are provided in Section 6.5.

6.1 Large Foil Offset Case

The purpose of this shot series was to determine the MRT growth rate for a foil displaced from the midplane. According to the ESM described in Section 3.2, the restoring $\mathbf{J} \times \mathbf{B}$ force would accelerate the foil plasma toward the midplane and supply

the necessary acceleration to drive the instability. Using the geometry described in Figure 3.1, where the load region was viewed from the same direction as observed from the laser backlighter images, the initial foil offset, x_0 , was negative, so the acceleration vector in the lab reference frame was positive. In the foil plasma's accelerating reference frame the equivalent gravity vector, \mathbf{g} , was negative. The stability of each interface was therefore determined by the Atwood number, given in Equation 3.110, according to the Rayleigh growth rate scaling presented in Equation 3.111. The vector \mathbf{g} may be interpreted as an equivalent gravity vector for the reference frame accelerating with the interface with the region 2 fluid above the region 1 fluid in this reference frame. The fluid containing only magnetic field was of negligible density compared to the foil plasma. The left side of the foil was a magnetic field \rightarrow plasma interface, so the Atwood number was positive and the interface was unstable. The right side was a plasma \rightarrow magnetic field interface, so the Atwood number was negative and the interface was stable. For a very thin foil, as determined by the parameter $k\Delta$, where k is the total wavenumber for the MRT instability and Δ is the foil thickness, the feed-through effect may strongly couple growth from an unstable interface onto a stable interface. This feed-through is 100% in the ESM model.

6.1.1 Initial Characterization

The initial position of the foil between the anodes was not perfectly reproducible due to imperfections in the load hardware, and due to the foil tensioning process that was performed with the experimental chamber under vacuum so that the approximately 0.5 cm of vertical compression was removed to straighten the foil. A preshot image of the foil load region is shown in Figure 6.1. The dark regions on either side of the interaction region are the anode plates, and the foil is the shadow-like filament backlit by the laser

pulse. The preshot images were crucial for determining the anode spacing, initial foil location and condition, and the initial locations of the left and right foil boundaries as viewed by the laser backlighter diagnostic.

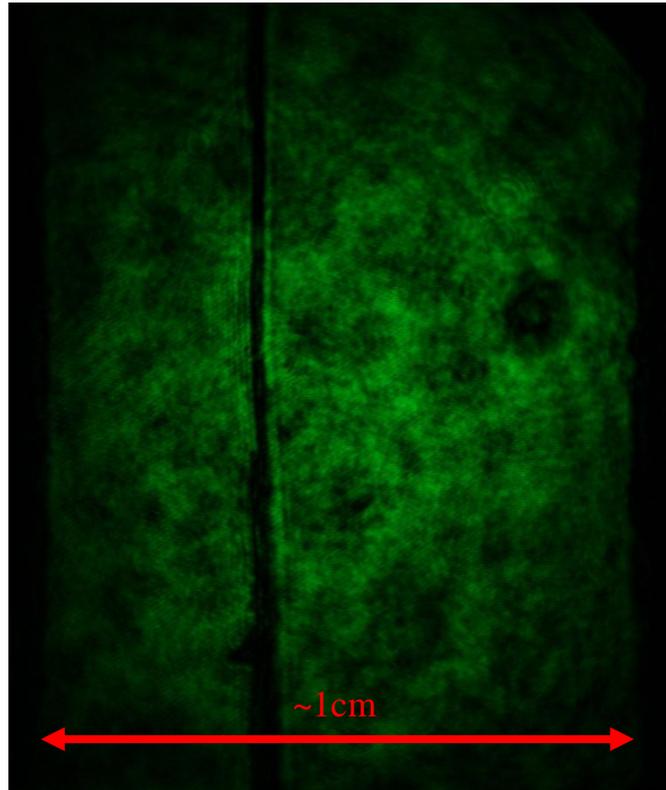


Figure 6.1 A preshot image of a foil load offset from the midplane. The prescribed anode spacing is shown in red.

A histogram of the initial foil positions is shown in Figure 6.2 (a). These positions were determined by measuring from the middle of the observable foil thickness to the midplane between the anodes. The observable foil thickness was the width of the shadow cast by the foil as the laser passed through the load during a preshot image. This is shown in Figure 6.2 (b). This width was primarily due to effects other than the actual foil thickness that, at 400 nm, was two orders of magnitude below the resolution of the imaging system as described in Section 4.3. These effects included the combination of any variability in the load cartridge placement relative to the laser axis, the small, but non

zero, 0.86° angles between the centerline and laser paths through the load, and due to diffraction effects that limited the resolution of the cameras. The small variability between the load centerline and the laser paths could not easily be decoupled without improvements in the cartridge hardware.

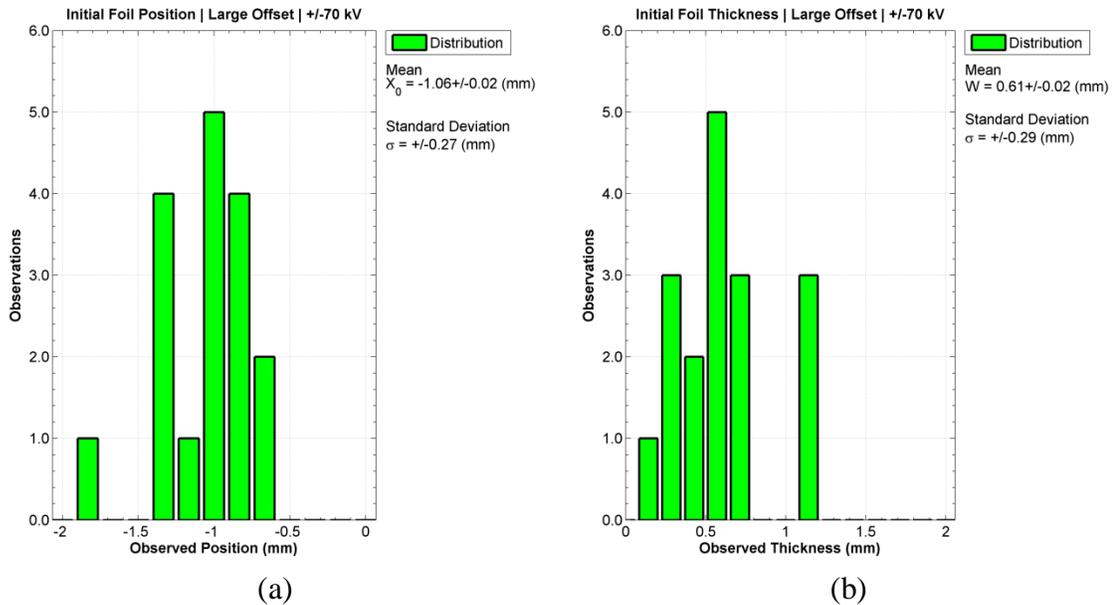


Figure 6.2 (a) The initial foil position distribution for the large offset case. (b) The initial observable foil thickness for the large offset case as viewed by the laser backlighter.

The mean initial position over the large offset case was -1.1 ± 0.3 mm. The variability in the initial position was partially due to the foil loading procedure, and partially due to the tensioning process. One anode piece was removed from the cartridge to load the foil for every large offset shot in order to gain access to clamping screws on the cathode. The second anode piece remained attached to the anode bridge. While great care was taken to reattach the removed anode in exactly the same position so that the foil's position between the anodes would be reproduced, this proved challenging with the hardware used, which relied on slotted channels rather than pinned attachment locations, and due to the sub millimeter precision required. Calipers could not be directly placed

between the anode plates because the support posts, present during the loading process to hold the cathode to the anode bridge, blocked access to the interaction region, and *in situ* adjustments after the posts were removed severely risked destroying the fragile foil. Tensioning also limited the foil placement reproducibility because the side edges of the narrow slot in the anode bridge initially contained, or may have developed over the shot campaign, a small ridge that could force the foil edges to one side or the other, effectively creating a small change in the foil position on the order of half a millimeter. If only one edge of the foil became caught by the ridge, a slight twist in the foil would result, which may have contributed to the 0.6 ± 0.3 mm mean observed thickness. Despite the need for future improvements in the load hardware, 17 quality images were obtained over 10 large offset shots to characterize the instability.

6.1.2 Temporal Evolution Observations

Figure 6.3 details the timings for the laser backlighter images superimposed onto their respective drive current pulses as measured by the B-dot probe as described in Chapter 4. A drive current threshold of 10 kA was set as the $t = 0$ point. This value was chosen because it was the lowest initial value that could always be distinguished from numerical noise as the beginning of the drive pulse. This convention ensured that a uniform start time was used for comparing image times between shots, and current flow prior to this point was of negligible magnitude and duration compared to the overall pulse maximum and risetime.

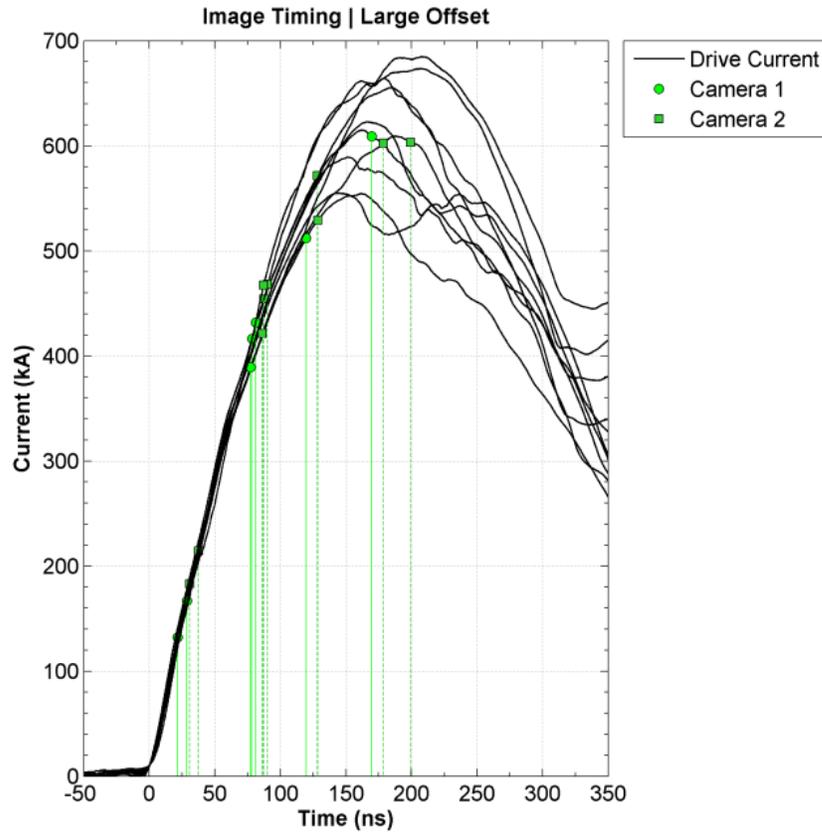


Figure 6.3 The image times plotted onto their corresponding drive current waveforms for the large offset case.

The drive currents were within $\pm 8\%$ or better until 100 ns into the pulse, after which they diverged by as much as 160 kA around peak current. The exact cause of this divergence was not well established, but may have been caused by changing load impedance due to the dynamic nature of the unstable foil load plasma. The fractional spread in current was approximately 18%. Assuming the product of the inductance and the current was approximately constant; this suggested an 18% fractional change in inductance from shot to shot, which could have occurred for a 0.4 mm variation in where the current was flowing through the expanded foil plasma shown in Figure 6.4 and Figure 6.5. The variations in drive current made shot-to-shot comparisons of backlighter images difficult at later times, particularly around peak current, however; the three images taken

around 600 kA were in the middle of the measured peak currents and represented the behavior for an average 600 kA current pulse. Between 325 ns and 350 ns the B-dot probe always began to produce unphysical signals by not integrating back to zero current, or by not tracking the pulse ringing predicted by the SPICE simulations shown in Chapter 5, instead remaining at some intermediate current between 200 kA and 450 kA. This suggested that the raw derivative signal from the inductive current monitor was prematurely terminated on dynamic load shots. While the exact nature of this termination was not well understood, the effect occurred well after peak current and after the laser diagnostic could no longer penetrate the plasma filling the load region during this late time. Similar B-dot signal failures routinely occur on other pulsed power machines, such as Sandia National Laboratories' Z-machine,⁵⁹ when firing dynamic loads. The image data set diagnosed the plasma well from initiation up to 200 ns.

The temporal evolution of the foil plasma is documented by annotated images in Figure 6.4 and is continued in Figure 6.5. The complete data set is provided in Appendix A. The foil load evolution may be described by two distinct phases: the initiation phase, during which foil ablation was the dominant process and the observable foil interfaces remained stationary; and the expansion phase, during which the observable interfaces began to expand outward. The latest image taken that only displayed negligible expansion was at 38 ns and marked the best estimate for the end of the initiation phase, after which the expansion could easily be measured. At 38 ns the edges already acquired a large apparent initial velocity. From the image recorded at 29 ns and 167 kA for the large offset case, the foil surfaces were observed ablating into plasma in a manner that appeared qualitatively similar to the ablation phase for a wire array z-pinch. Finger-like

ablation streams were observed forming on either side of the foil, similar to those observed from ablating wires.³¹ These ablation streamers were clearly visible in Figure 6.4 (b). Lineouts from this image parallel to the left side of the foil are shown in Figure 6.6 along with a spatial Fourier transform showing the wavelength structure of the streamers. The extraction method for the lineouts, along with a description of the quoted threshold values, is provided in Section 6.3. The streamers appeared at approximately a 0.4 mm wavelength. This was similar to, but still distinct from the 0.5 mm wavelength typical for aluminum wire arrays. The difference may be due to the different magnetic topology. The ablation streams also appeared to travel toward the midplane more quickly than toward the left anode, as should be expected from the initial net magnetic pressure resulting from the offset position, although this preferential travel was significantly milder when compared to the ablation streams that form precursor columns on the central axis in a cylindrical wire array load.²⁶ This milder streamer motion may have resulted from at most half of the global magnetic field being available to propel the streamers, but in a wire array the entire global field is available to propel the streamers.

The most striking behavior observed was the expansion of a bulk plasma region from the foil that was inaccessible with the laser backlighter diagnostic. This inaccessibility could have resulted from: 1) plasma density regions above the critical density for 532 nm light, defined as the density above which the refractive index of the medium goes to zero, 2) from light passing through this region that was refracted away from the camera sensor, or 3) from the plasma having an optical depth that was too large for the camera sensor to resolve from the given laser energy, particularly when compared to the background self emission light levels from the plasma itself that penetrated the 532

nm bandpass filters. Self emission levels typically consumed 40% of the sensor's dynamic range, determined by taking the nearly zero signal level behind the anodes far from the load region on preshot images and comparing those preshot values to emission levels in the same region on data images where the laser was mistimed and fired before the start of current flow through the load.

To determine the critical density for the plasma medium at 532 nm, the dispersion relation for the laser light passing through the plasma must be known. This treatment neglects collisions, which is accurate further from the ablating foil in the interaction region. Close to the foil, opacity effects were expected to dominate as discussed below. Useful parameters for describing the plasma are the plasma frequency

$$\omega_p^2 = \frac{n_e e_0^2}{\epsilon_0 m_e}, \quad 6.1$$

where n_e is the electron number density, e_0 is the electron charge, ϵ_0 is the permittivity, and m_e is the electron mass; and the electron cyclotron frequency given by

$$\omega_c = \frac{e_0 B}{m_e}, \quad 6.2$$

where B is the magnetic field. The linearly polarized laser passed through the load region parallel to the major component of the magnetic field. This propagation and field geometry obeyed the L- and R-wave dispersion relation given by

$$\eta^2 = \frac{c^2 k^2}{\omega^2} = 1 - \frac{\omega_p^2 / \omega^2}{1 \pm \omega_c / \omega}, \quad 6.3$$

where η is the refractive index, c is the speed of light, ω is the angular frequency of the laser plane wave, and k is the wave number for the laser plane wave.⁶⁰ From Section 5.3,

the largest predicted magnetic fields were on the order of 86 T. Calculating ω_c using a 100 T magnetic field and comparing this frequency to the laser frequency, $\omega_c/\omega = 1.76*10^{13}$ (rad/s) / $3.54*10^{15}$ (rad/s) = $5*10^{-3}$, the ratio was found to be much less than unity, even for a magnetic field greater than the predicted value. This allowed Equation 6.3 to be simplified to the ordinary wave dispersion relation

$$\eta^2 = \frac{c^2 k^2}{\omega^2} = 1 - \frac{\omega_p^2}{\omega^2}. \quad 6.4$$

The critical electron density, n_c , occurs when $\eta = 0$ and is expressed as

$$n_c = \frac{\omega^2 \epsilon_0 m_e}{e_0^2}. \quad 6.5$$

The calculated critical density for the 532 nm laser light was therefore $3.94*10^{21}$ cm⁻³.

This high critical density was unlikely to be reached except very close to the ablating foil core in these experiments as discussed below.

To explore the refractive effects of the plasma on the backlighter beam, the plasma geometry was approximated as a slab of uniform plasma with an unknown electron density. By matrix multiplication, the transfer matrix equation for paraxial rays is given by

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 & d'_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -f^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & d_o \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \eta_p \end{bmatrix} \begin{bmatrix} 1 & d_p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \eta_p^{-1} \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}, \quad 6.6$$

for the optical path described by an initial ray with distance x_1 and angle θ_1 between the optical axis and the plasma slab and that traverses the following path: crossing the vacuum \rightarrow plasma interface, propagating within the plasma for a distance d_p , crossing the plasma \rightarrow vacuum interface, propagating through freespace a distance d_o , passing

through an imaging lens with focal length f , and propagating through freespace a distance d_i' to the viewing plane.⁶¹ This viewing plane was 7.6 cm in front of the actual camera sensor and represented the final aperture that the ray must pass through to be detected. Blockage at this aperture therefore represented the minimum deflection required to prevent detection. This aperture had a radius of 1.27 cm and was placed at a location x_2 such that when there was zero plasma and $\eta_p = 1$, the ray passed through the center of the aperture. This was equivalent to the preshot laser being aligned onto the detector through this aperture and allowed the turning mirrors and other optical path details to be calibrated out from the transfer matrix analysis, leaving only the elements important for the deflection calculation. The change in x_2 caused by the electron density in the plasma was then computed. This change needed to deflect the ray 1.27 cm for the aperture to block the light. The parameters for the calculation are provided in Table 2. The required electron density to produce the necessary deflection was calculated to be $3.94 \cdot 10^{21} \text{ cm}^{-3}$, just below the critical density.

Parameter	Value
x_1	0 (m)
θ_1	0.015 (rad)
d_p	1 (cm)
d_o	1.854 (m)
f	1 (m)
d_i'	2.108 (m)
$\eta_p (n_e)$	$\sqrt{1 - 2.539 \cdot 10^{-28} [m^3] * n_e [m^{-3}]}$

For the assumed uniform plasma slab geometry used for this analysis, refraction was clearly only appreciable near the critical density. The real plasma was of course more complicated than a uniform slab, certainly containing internal density gradients that

would require path integration to find the total deflection, but that detailed analysis is beyond the scope of a simple analytic estimate and would require post processing on MHD simulation results to generate accurate synthetic images to determine the exact role of refraction in producing the inaccessible bulk region observed in the data.

The same complexity limited analytic estimates for the optical depth of the foil plasma because the ionization populations, interaction cross sections, and internal density profile for this plasma quickly became too complex for a simple calculation. Again, synthetic image generation from simulations would have been required to obtain accurate measurements for the role plasma opacity played in producing the bulk region. What could be qualitatively stated was that large deflections required densities approaching the critical density, and that if a uniform density distribution that assumed 100% of the initial foil material had ablated and expanded to a 1 mm width, the number density would only be $2.4 \times 10^{19} \text{ cm}^{-3}$. This overestimated the density at the edges of the bulk region because it was unlikely that the entire foil ablated when a 1 mm expansion would have been a reasonable size, based on ablation patterns for z-pinch wires where dense cores remain unablated inside coronal plasma well into the drive.²⁴ Additionally, the density profile intuitively should have been peaked at the center, around the remaining foil core, and then decreased toward the edges. This overestimate for the density, even if 100% ionization was assumed to increase the value by an order of magnitude, was still an order of magnitude below the critical density for 532 nm light. So while cutoff and refraction certainly played a role, particularly close to the presumed foil core, it seemed qualitatively reasonable to attribute the outer edges of the bulk region to be a result of optical depth, meaning there was simply too much absorption by material inside this

region for the laser to penetrate and be detected by the camera sensor. This analysis also did not rule out the integrated effect from many small angle refractions resulting from severe density gradients as the laser traversed the plasma. Again, the detailed density gradient structure within the plasma was not available, and although schlieren techniques sensitive to density gradients could probe the extreme edges of the bulk plasma,⁶² the contribution from these gradients still could not be easily decoupled from opacity.

An instability appeared to develop on both the left and right sides of the foil beneath the 0.4 mm ablation structure. This is clearly illustrated by shot 306 in Figure 6.5 (e) at 120 ns. While the wavelength for this structure was not as uniform as for the ablation streamers, it appeared to have wavelength modes at approximately 0.8 mm and 1.2 mm. This wavelength measurement required estimates from lineouts such as Figure 6.6 (c), rather than a direct measurement from the spatial Fourier transform (d), because the transforms did not have adequate resolution for wavelengths longer than roughly 0.5 mm, although the transforms did indicate that there was a strong wavelength between 1 mm and 1.5 mm. This resolution was a limitation set by the physical length of the lineout, which was maximized, but ultimately fixed by the height of the load interaction region. This longer wavelength structure was believed to be the MRT instability developing on the interface between the bulk plasma region and the surrounding magnetic field. It was very interesting that the instability was observed on both sides of the foil instead of solely on the left side, as predicted from the ESM, Rayleigh scaling, and the Atwood number for each interface.

One additional feature that began to develop during the 81 ns image was what appeared to be anode plasma expanding from both current return electrodes. By 179 ns

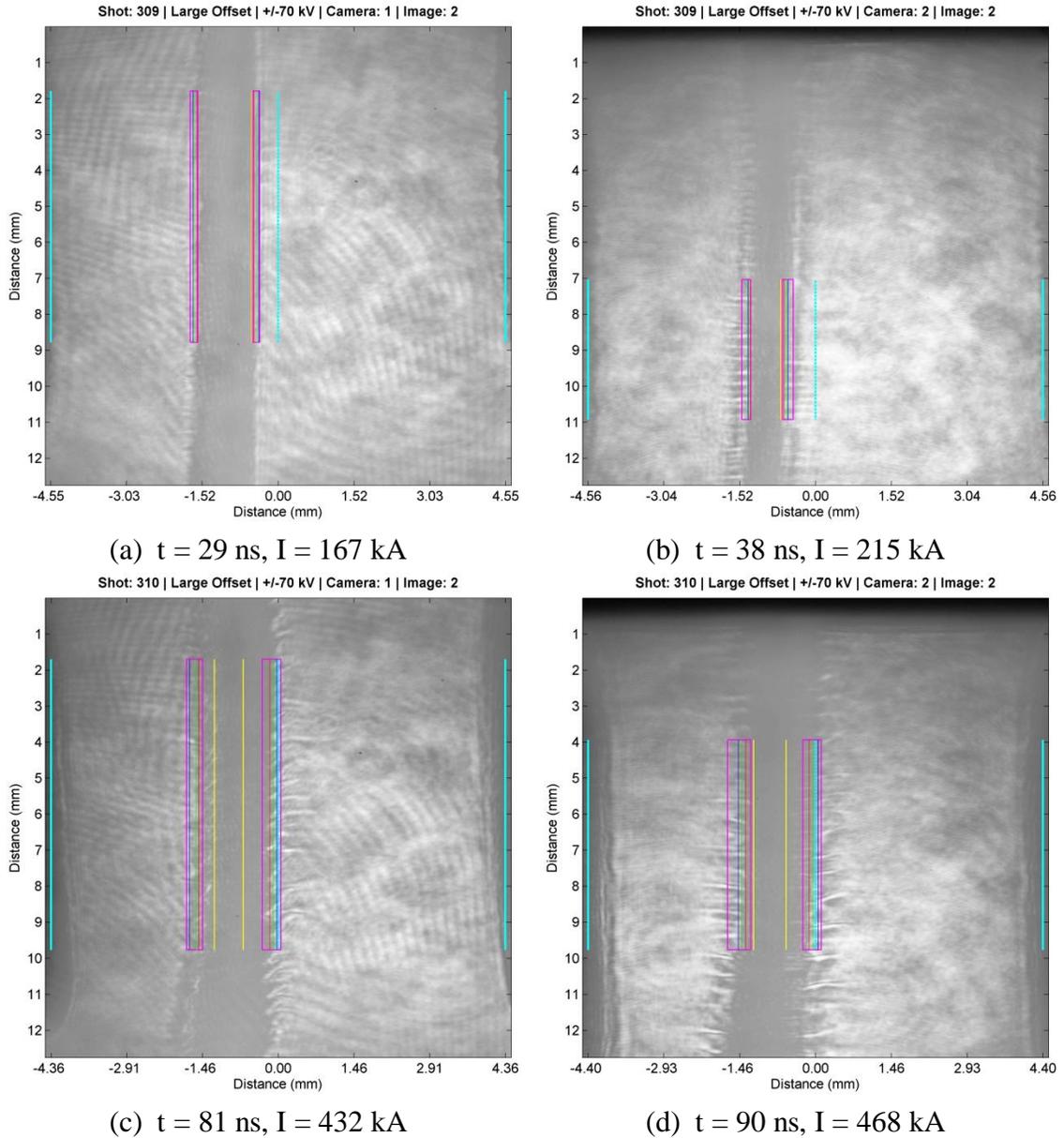


Figure 6.4 The temporal evolution of the foil plasma for the large offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.

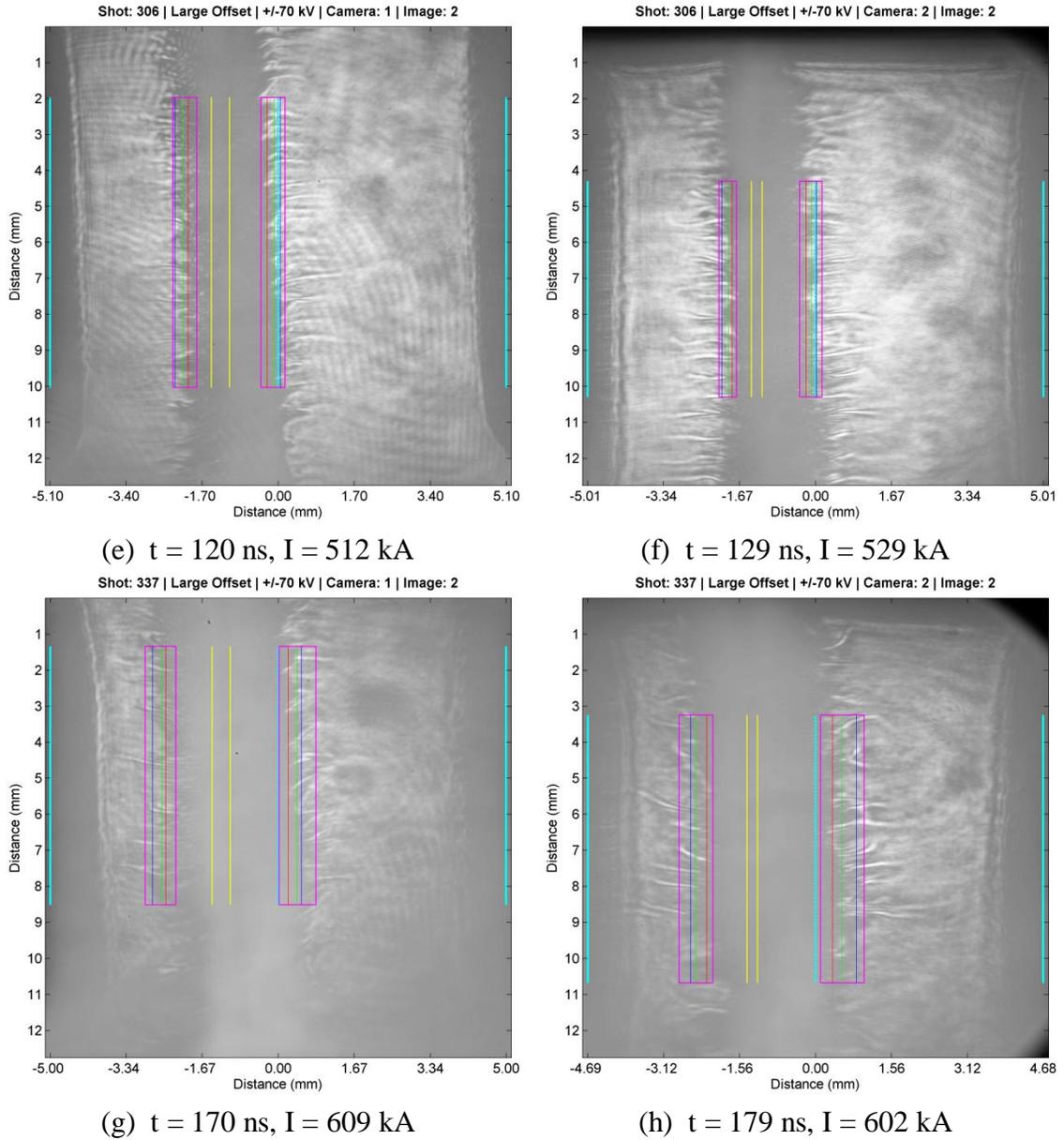


Figure 6.5 The continued temporal evolution of the foil plasma for the large offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after than camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.

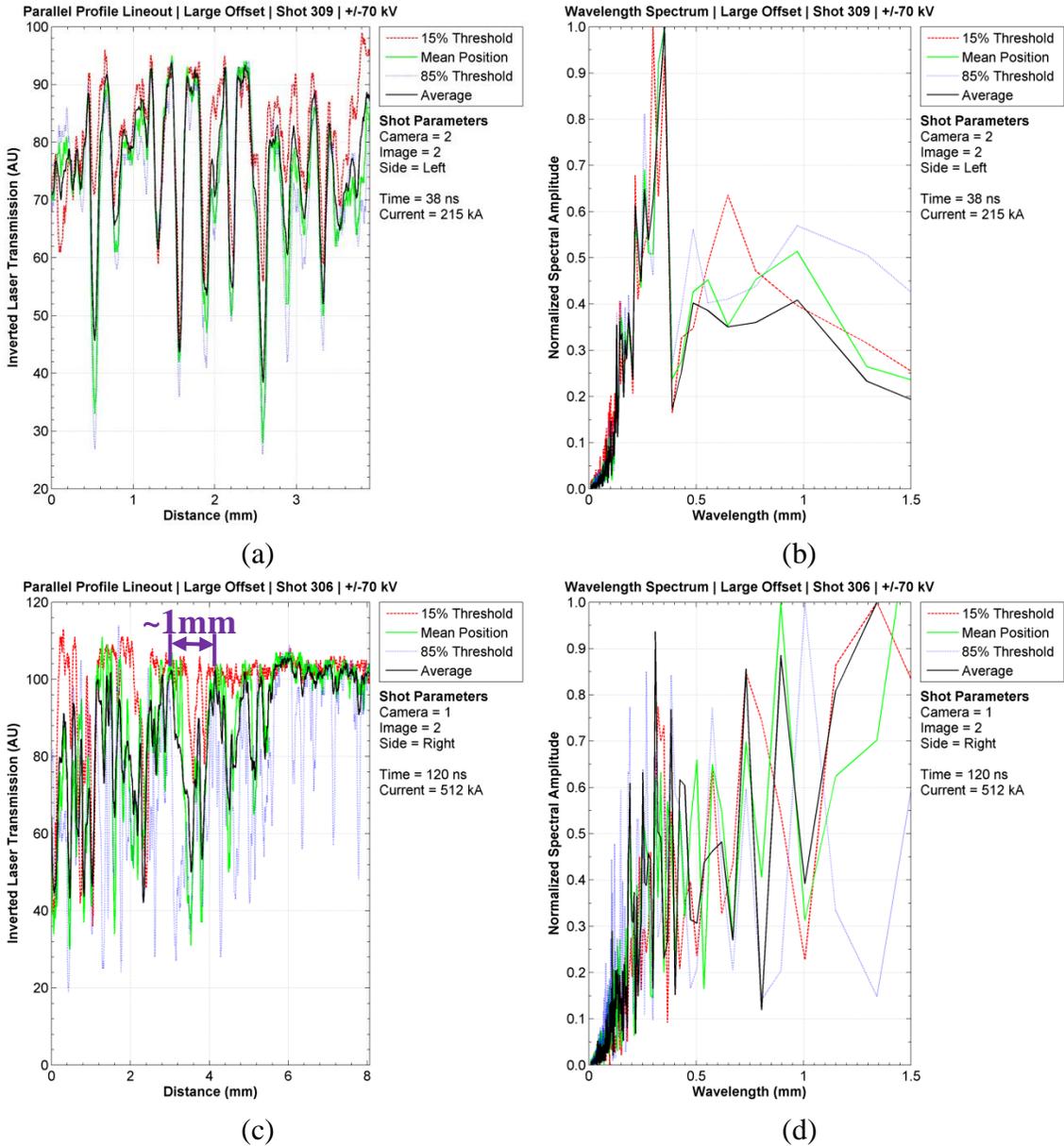


Figure 6.6 (a) Lineouts extracted parallel to the left side of the foil from shot 309. The ablation fingers create the oscillatory density pattern. (b) The Fourier transform of part (a) showing the strong 0.4 mm wavelength. (c) Lineouts extracted parallel to the right side of the foil from shot 306. The ablation structure appears superimposed onto an enveloping structure with approximately a 1 mm wavelength. (d) The Fourier transform of part (c).

this feature had expanded over 1 mm from the original electrode position. The plasma was presumably created by the roughly 300 kA/side peak return current depositing energy into the skin depth surface of the anodes. This anode plasma appeared optically thick to the laser and thwarted imaging attempts significantly beyond peak current. It was unclear what shorting effect this gap closure had later in time because the B-dot monitors failed after 350 ns. The average expansion calculated from Figure 6.5 (h), $v_{ap} = 1.17 \text{ mm} / 179 \text{ ns} = 0.65 \text{ cm}/\mu\text{s}$, would close off roughly $1/4^{\text{th}}$ of the interaction region at 350 ns, which suggested that the combination of anode plasma expansion meeting the foil plasma from both sides may have occurred around the time the B-dot monitor failed, but this may also have been a simple coincidence.

6.2 Small Foil Offset Case

The purpose of this shot series was to determine the MRT growth rate for a foil placed as close as feasible to the midplane such that the foil would theoretically experience a zero net $\mathbf{J} \times \mathbf{B}$ force according to the ESM in Section 3.2. In the absence of acceleration, both interfaces should have experienced zero MRT instability growth.

6.2.1 Initial Characterization

The initial position of the foil between the anodes was again not perfectly reproducible for the same reasons described in Section 6.1.1 for the large offset case. A preshot image of the foil load region is shown in Figure 6.7 where the foil was placed as close as possible to the midplane. The positioning reproducibility was improved during the small offset shot series by loading the foils into the cartridge assembly without removing one of the anode plates. While it was more difficult to load the foil in this

manner, the foil position inside the cartridge relative to the anode plates was no longer influenced by small anode attachment inconsistencies.

A histogram of the initial foil positions for the small offset case is shown in Figure 6.8 (a), and the observable foil thicknesses are shown in Figure 6.8 (b) based on preshot images. The mean initial position over the small offset case was 0.1 ± 0.3 mm. The

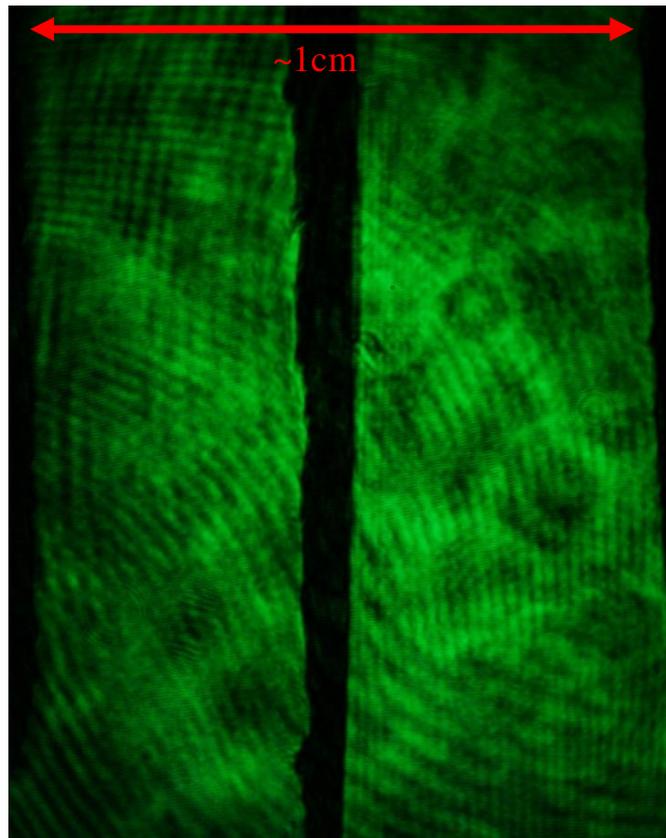


Figure 6.7 A sample preshot image for the small offset case. The prescribed anode spacing is shown in red.

standard deviation was identical to the large offset case, although the improved foil loading procedure was most likely responsible for the 9 measurements within a single 0.13 mm wide histogram bin near zero. The observed initial thickness was 0.8 ± 0.3 mm, which contained a mean thickness 0.2 mm larger than that measured from the large offset case, although both means were within a standard deviation of each other. It was unclear

why the observed thickness increased. Thirty quality laser images were obtained over 15 small offset shots to characterize the instability for this case.

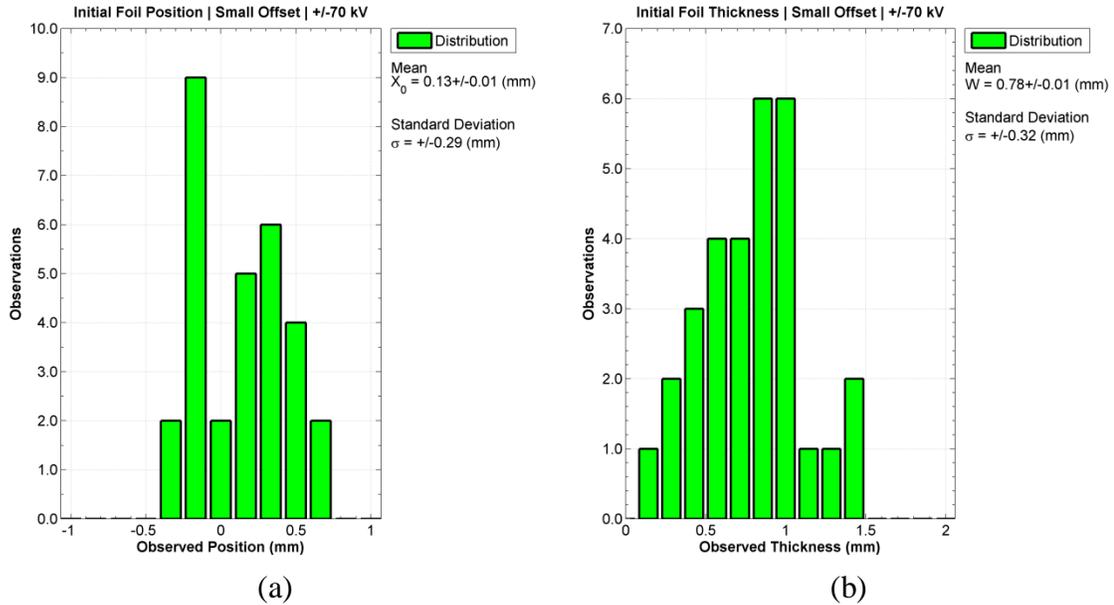


Figure 6.8 (a) The initial foil position distribution for the small offset case. (b) The initial observable foil thickness for the small offset case as viewed by the laser backlighter.

6.2.2 Temporal Evolution Observations

Figure 6.9 details the timings for the laser backlighter images superimposed onto their respective drive current pulses as measured by a B-dot probe. As with the large offset case, a current threshold of 10 kA was set as the $t = 0$ point. The drive currents were within $\pm 4\%$ or better until the majority of pulses reached a peak current of 625 kA after 160 ns. These currents were significantly more uniform than those shown in Figure 6.3, which may have been a result of the more symmetric load, and suggested that load dynamics were indeed a significant contributor toward the variations near peak drive current observed in the large offset case. The plasma was well diagnosed over the current risetime up to 179 ns.

The temporal evolution of the foil plasma is documented by annotated images in Figure 6.10 and is continued in Figure 6.11. The complete data set is provided in Appendix A. The most noticeable and striking observation from the small offset images was how similar they appeared to those from the large offset case. The foil began to appreciably ablate between the 7 ns image and the 16 ns image, forming ablation streamers with a wavelength between 0.3 mm and 0.5 mm. The length of the streamers appeared to be roughly symmetrical between the right and left sides, whereas the streamers in the large offset case images tended to be longer on the right side than on the left. The foil plasma entered the expansion phase and expanded outward about its initial position, and the same MRT instabilities developed on both sides of the foil. The

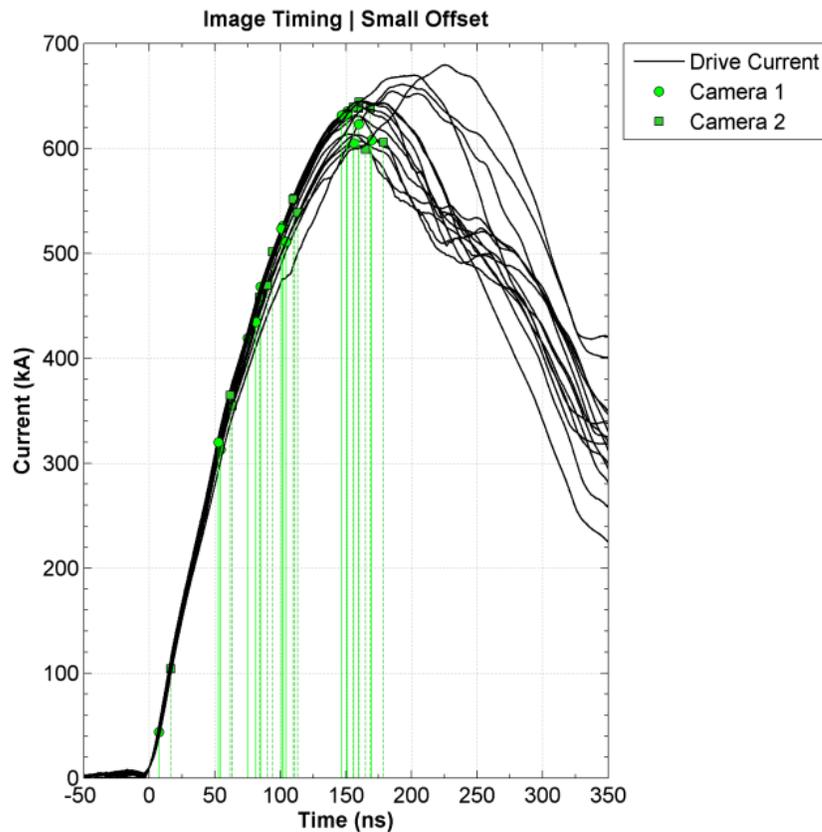


Figure 6.9 The image times plotted onto their corresponding drive current waveforms for the small offset case.

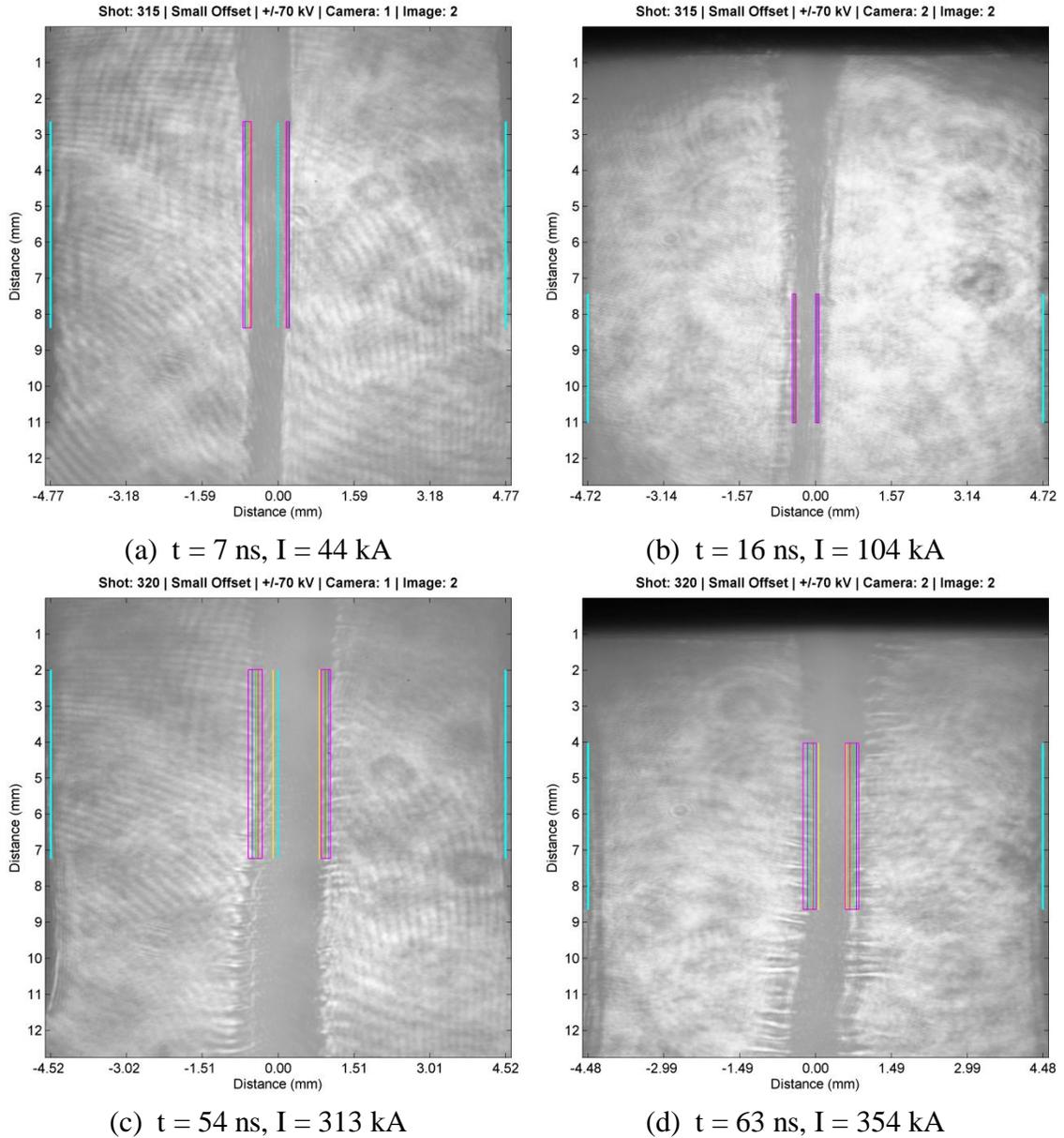
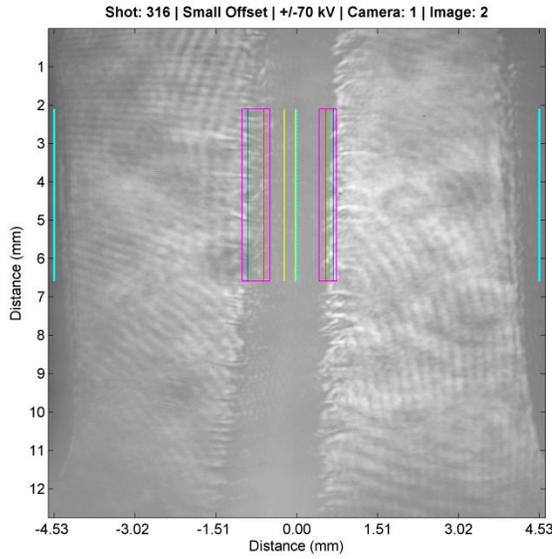
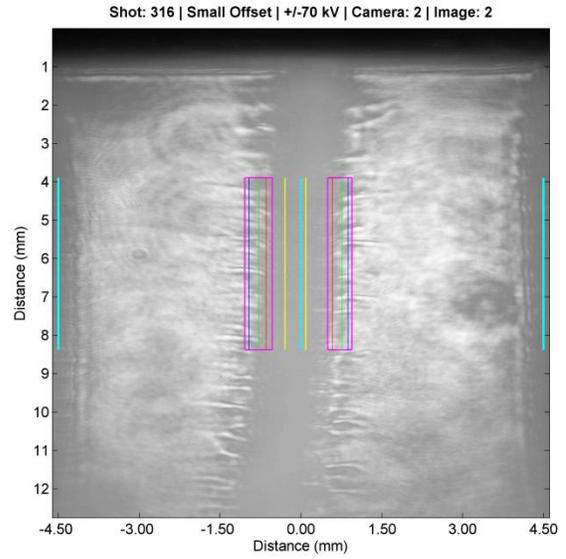


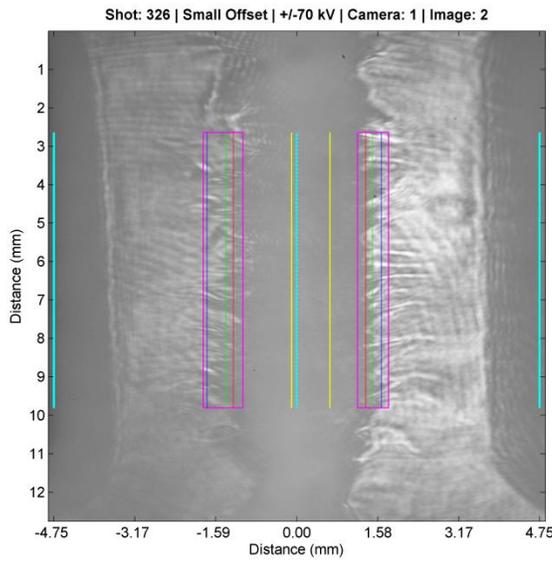
Figure 6.10 The temporal evolution of the foil plasma for the small offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.



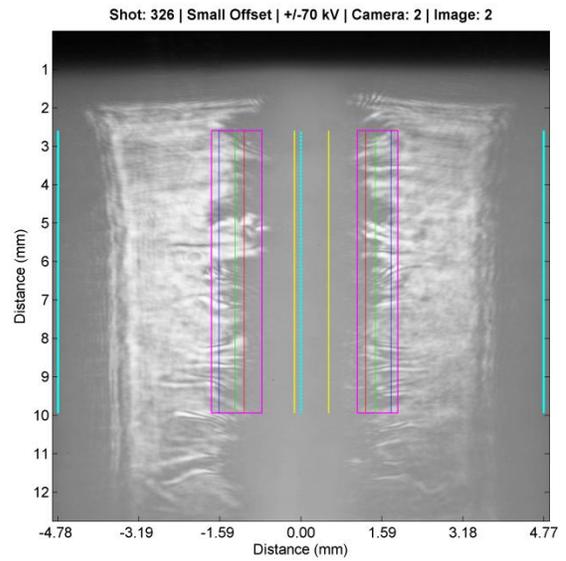
(e) $t = 85$ ns, $I = 468$ kA



(f) $t = 94$ ns, $I = 501$ kA



(g) $t = 170$ ns, $I = 608$ kA



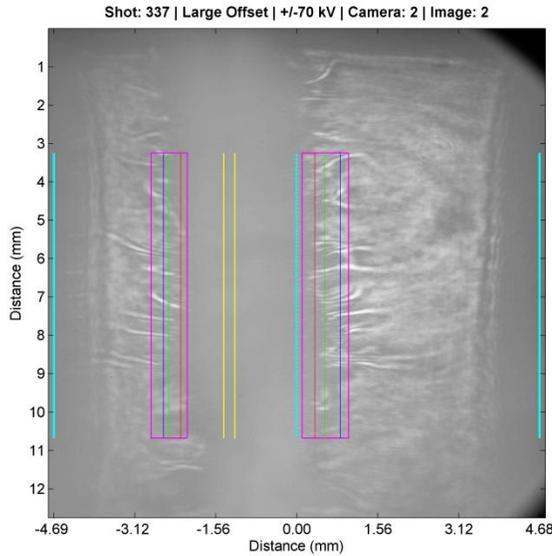
(h) $t = 179$ ns, $I = 606$ kA

Figure 6.11 The continued temporal evolution of the foil plasma for the small offset case. The time and drive current are shown below each image. Images on the left side are from camera one for a given shot, whereas images on the right side are from camera two, 9 ns after camera one on the same shot. The positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.

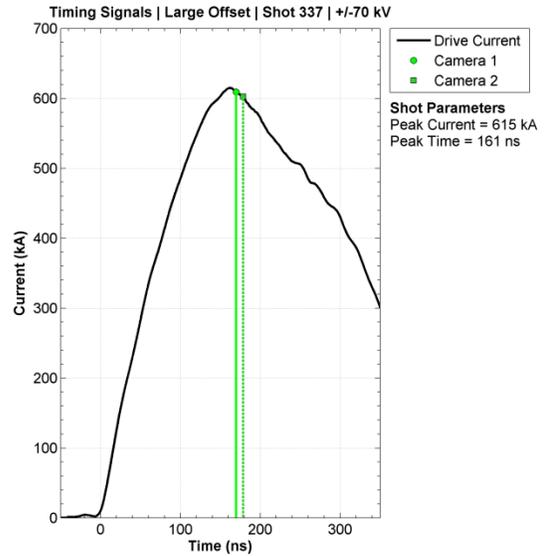
wavelength for the instability appeared to have a strong 0.8 mm mode, and perhaps the longer 1.0 mm to 1.2 mm mode, although detailed differences for these longer wavelengths were again difficult to discern due to the limited Fourier transform resolution and the multimoded nature of the lineouts.

6.3 Quantitative Image Analysis

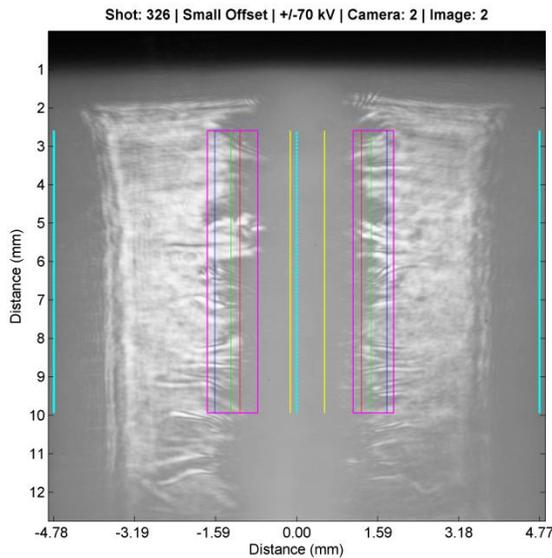
The data sets for both cases were analyzed using a custom suite of Matlab programs to process the raw data files. Information for each shot was stored in a parameter file that the Matlab suite accepted as an input, which allowed data from additional shots to be automatically analyzed simply by updating the parameter file to include the new information. A user control panel inside the Matlab suite allowed the various analysis options to be enabled or disabled as desired. These codes are provided in Appendix B. The most important parameters for analyzing a shot were the control coordinates for each image. The color codes for the following features refer to the annotation lines found in Figure 6.12 and in all other annotated data image figures. These consisted of pixel locations from both the preshot image and from the data image. The preshot y-coordinates were chosen to be approximately 2 mm or more away from the top and bottom of an image to avoid electrode effects or changing anode spacing due to the fillets on the bottom of the anode return plates, and were otherwise as long as possible to maximize the analyzed foil height. The analyzed height and location were also chosen based on how straight and thin the observable foil thickness was on a given preshot image. The thinnest, tallest, and most uniform region was chosen. If the corresponding location on the data image contained unanalyzable regions due to poor contrast, the height was adjusted to exclude such a region. The preshot x-coordinates included the



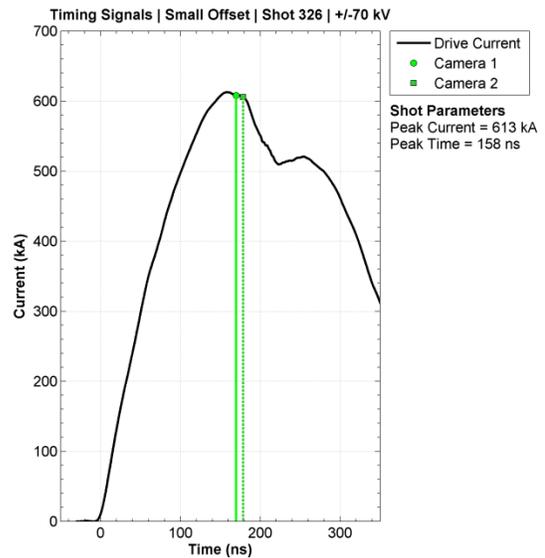
(a) $t = 179$ ns, $I = 602$ kA



(b)



(c) $t = 179$ ns, $I = 606$ kA



(d)

Figure 6.12 (a) An annotated image and (b) corresponding current drive plot for a large offset case shot. (c) An annotated image and (d) current for a small offset case shot at the same time and current. In (a) and (c), the positions, taken from preshot images, of the anode plates and the observable foil boundaries are shown in cyan and yellow respectively. The regions of interest for the instability interfaces are shown in magenta, and the 15%, mean, and 85% transmission thresholds are shown in red, green, and blue respectively. The images have been stretched horizontally by approximately 30% to aid the reader with viewing the instability development.

initial anode boundaries, shown in solid cyan, and the initial observable foil boundaries, shown in yellow, within the y-coordinate locations. The midplane, shown in dotted cyan, was calculated from the anode boundaries. The data image locations consisted of two x-coordinates on each side of the foil. These, along with the y-coordinates, defined the region of interest (ROI), shown as magenta boxes, to extract and analyze. These values were very carefully chosen and repeatedly checked to ensure that the correct regions from the data image were being extracted. The locations were chosen as the best possible judgments by eye for the amplitude of the MRT instability developing beneath the ablation streamers on the bulk plasma interfaces. The ROIs therefore excluded the vast majority of the streamer opacity until MRT growth modulated those later in time. After modulation, this opacity of included in the ROIs. Once the ROIs for a given image were determined, additional measurements within each ROI were performed; perpendicular to the foil surface, meaning the horizontal direction; and parallel to the foil surface, meaning the vertical direction. For most images the ROI boundary locations were obvious, although determining them for the earlier times prior to the expansion phase was more difficult due to the presence of the ablation flair structure that dominated during the initiation phase. Since the instability being measured occurred on the interface with the bulk plasma, and that prior to expansion this region was the initial observable foil location, the early time ROIs were consequently a combination of whatever bulk plasma had formed, along with the initial bumps on the foil itself.

6.3.1 Perpendicular Laser Transmission Profile Lineouts

Perpendicular lineouts were generated from the ROIs by averaging the ROI laser transmission values vertically, and then plotting those averages as a function of

horizontal distance away from the foil. This analysis perpendicular to the foil was a useful diagnostic for characterizing the mass distribution, even for an instability that contained many wavelengths. These perpendicular transmission profiles were, in general, monotonically decreasing on the left side, and increasing on the right side. The

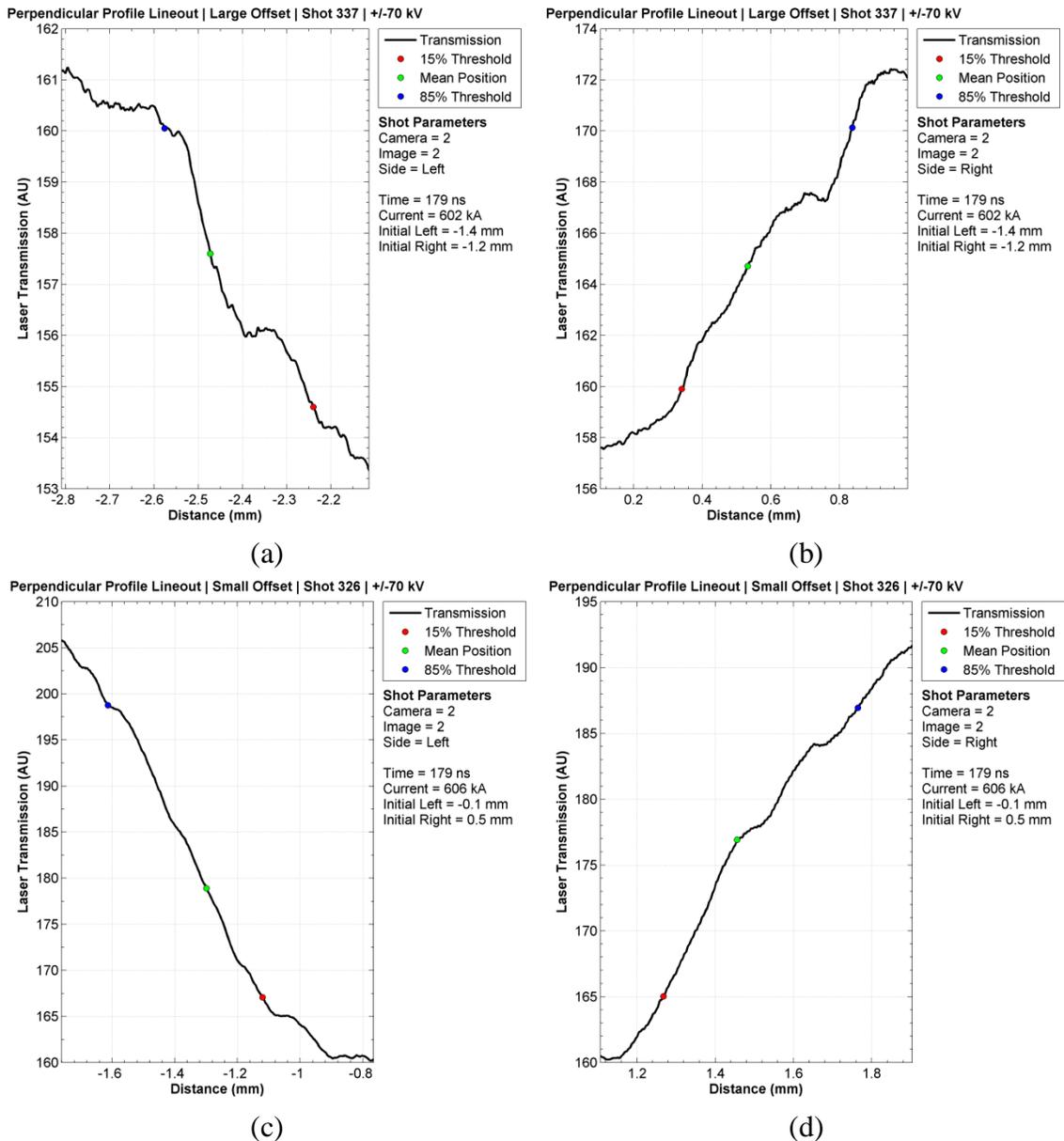


Figure 6.13 (a) Example perpendicular lineouts from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side lineouts for a small offset shot.

perpendicular lineouts corresponding to the data images in Figure 6.12 are shown in Figure 6.13. Three laser transmission threshold levels were chosen from these perpendicular lineouts: a 15% level, meaning 15% of the laser transmission range within the lineout; an 85% level, calculated in the same manner; and a mean level, meaning the average transmission value within the lineout. The corresponding distances for each threshold value were then recorded to characterize the height of the instability structure. The 15% threshold tracked the structure close to the bulk plasma, whereas the 85% threshold tracked the material further away. The greatest challenge for determining these threshold levels was the presence of diffraction halos, particularly on early time images during the initiation phase. These halos would generate a severe local maximum and minimum within the lineout, forcing the lineout to become multivalued for a given threshold, particularly for the mean. This was addressed by approximating where the mean value would have occurred without a diffraction halo by taking the midpoint between the first and last points in the lineout at the mean threshold. This very effectively located the central mean value between the local maximum and minimum. The 15% and 85% thresholds happened to be too low and high respectively to become multivalued due to diffraction halos for the given data images.

6.3.2 Parallel Laser Transmission Profile Lineouts

Information regarding the mode structure for the instability was obtained using lineouts parallel to the foil. Four parallel lineouts were plotted for each side. Three of these were lineouts taken at the threshold values determined from the perpendicular lineouts. Additionally, a fourth lineout was generated by horizontally averaging the ROI between the 15% and 85% threshold distances, and then plotting those averages against

vertical distance along the foil. The corresponding parallel lineouts from the data images in Figure 6.12 are shown in Figure 6.14. These profiles were inverted to show opacity, rather than transmission, as a peak.

6.3.3 Wavelengths of Transmission Profiles

The parallel profiles were also analyzed in the spatial frequency domain by performing a spatial Fourier transform on the lineouts as shown in Figure 6.15. These spatial frequency spectra were converted to wavelength and normalized. The resolution for most spectra was sufficient below 0.5 mm to identify specific wavelengths and improved with the vertical length of the original ROI. Above this value the data points became sparse and it became difficult to determine a wavelength precisely. Typical approximate peak locations across the dataset included 0.4 mm, likely due to the ablation streamers, 0.8 mm, and 1.2 mm. The longer wavelengths were most likely the modes for the MRT instability, although a much longer foil would have been required to achieve sufficient resolution in this wavelength range to clearly identify the modes and to track their evolution through time.

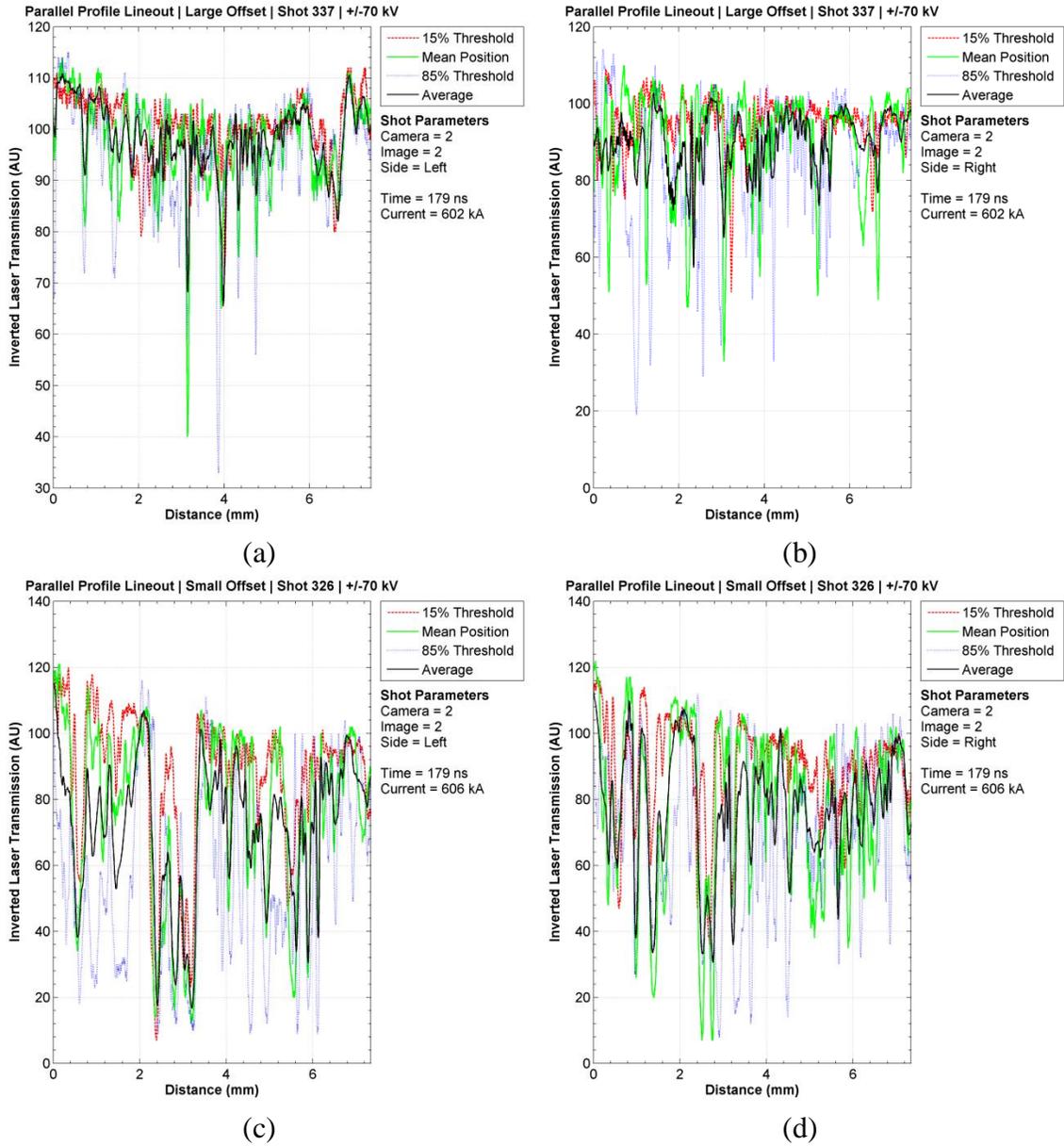


Figure 6.14 (a) Example parallel lineouts from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side parallel lineouts for a small offset shot. The distances were relative to the top of the ROI.

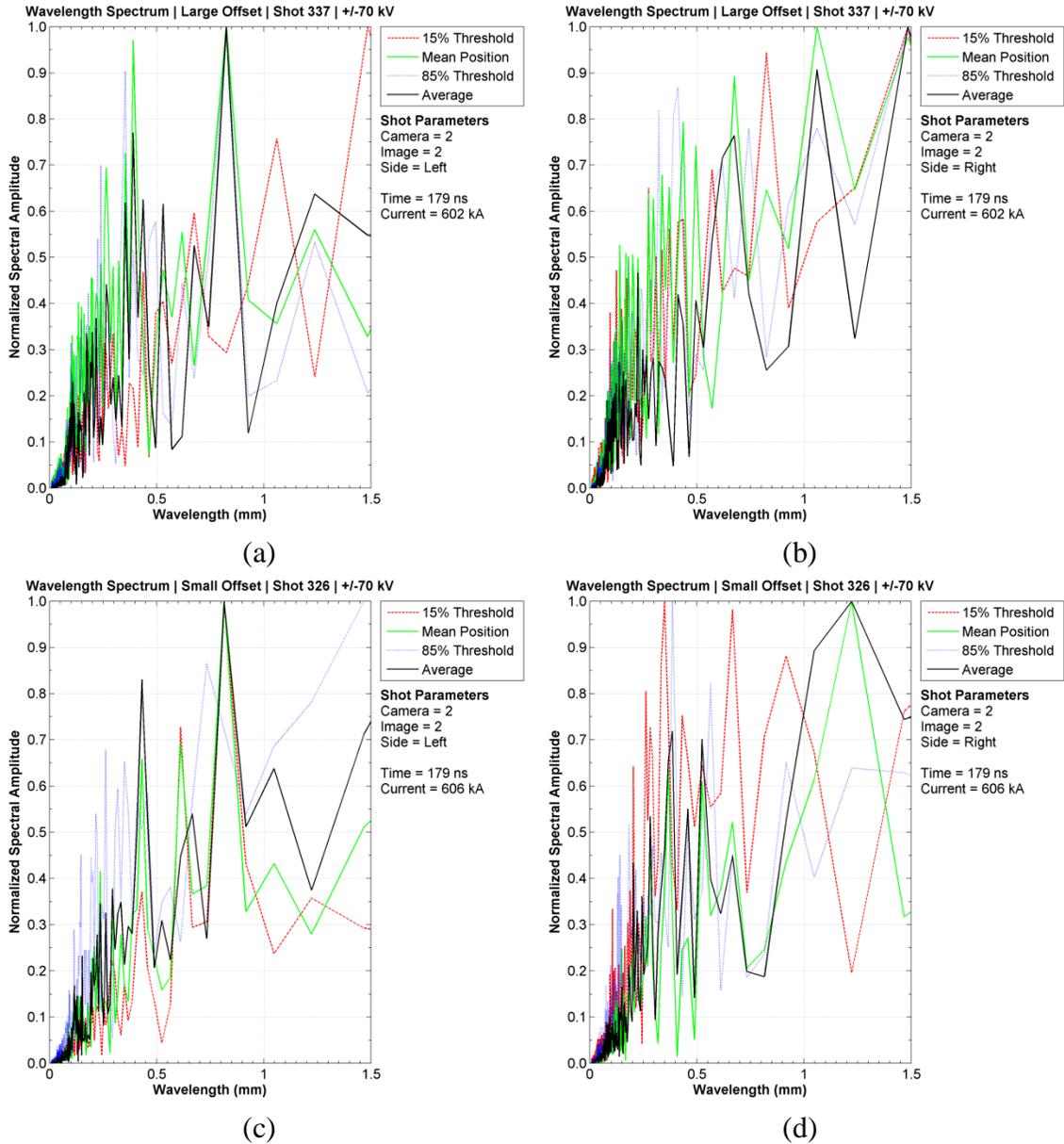
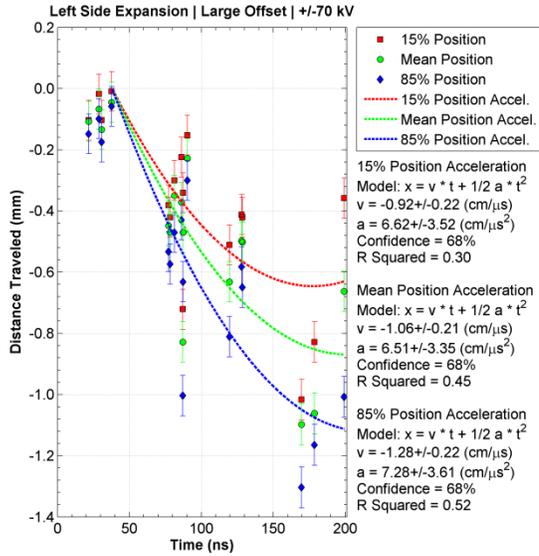


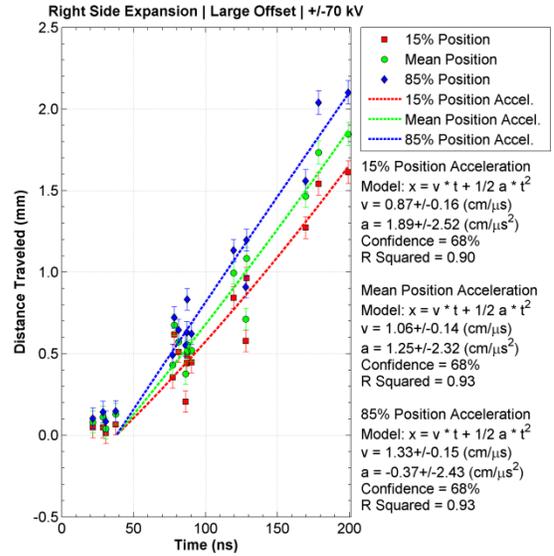
Figure 6.15 (a) Example parallel lineout Fourier transform spectra from the left side and (b) right side of a large offset shot. (c) The corresponding left side and (d) right side parallel wavelength spectra for a small offset shot. The spectra were individually normalized.

6.4 Foil Plasma Expansion

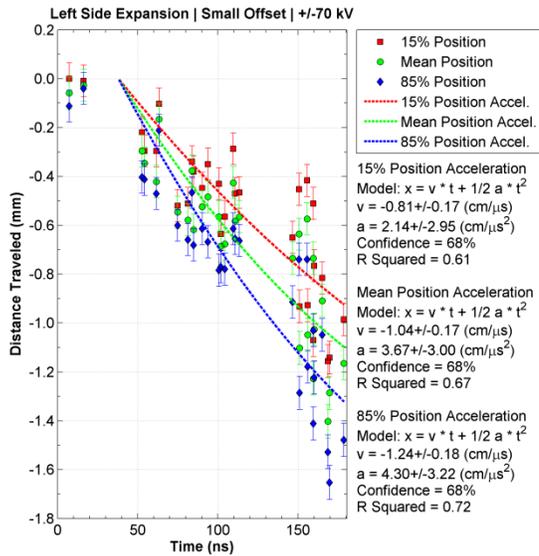
The transmission threshold locations were recorded on each side of the foil for both data sets. The initial observable foil location on a given side was subtracted to obtain a metric for the change in the threshold locations as a function of time. The threshold locations were needed to track the movement because the raw interface was a multi-moded structure that required opacity weighting to characterize the amplitude. These measurements are shown in Figure 6.16 for both sides of the foil and for both the large and small offset cases. Since the positive x-direction was defined as a displacement to the right, expansions to the left that were characteristic of the left interface are shown as negative. The expansion measurements were empirically fit using a nonlinear least squares method to a parabolic model. This model assumed a delay time between the $t = 0$ point and the beginning of the expansion phase of 38 ns, the earliest time when expansion was measured. The fit model included an initial velocity coefficient and a constant acceleration coefficient. The data points within the delay time during the initiation phase were more a characterization of the observable foil surface rather than metrics for expansion because the ROIs during this time contained an unknown combination of foil material at the initial location, as well as the ablating plasma that would eventually expand outward. These initiation phase data points were not included in the fits. The exceptionally large uncertainties originated from the requirement to fit both the velocity and acceleration coefficients at the same time, both assumed to be constant throughout the drive current risetime. These uncertainties and idealizations severely limited the use of these coefficients for detailed quantitative measurements.



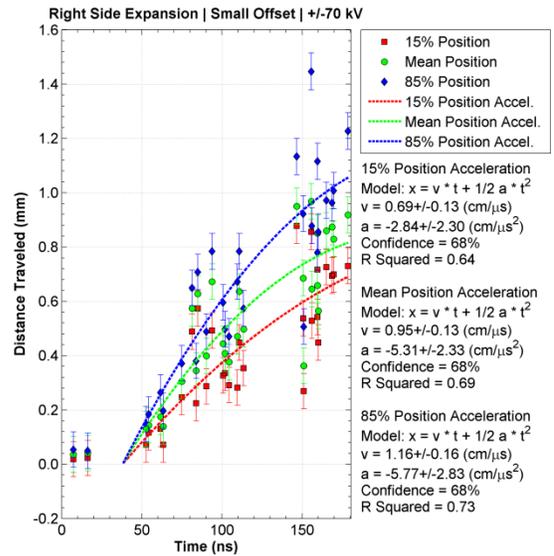
(a)



(b)



(c)



(d)

Figure 6.16 Expansion measurements referenced to the initial observable foil position for (a) the left side of the large offset case, (b) the right side for the large offset case, (c) the left side for the small offset case, and (d) the right side for the small offset case. Empirical best fit curves are shown for the expansion phase with a 38 ns delay time.

The initial velocity of the bulk interface may have been serving as a metric for a combination of more complicated physical effects during the initiation phase that resulted in the interface having what appeared to be an initial velocity to the laser diagnostic. One possible physical effect was ablated material being blown outward by kinetic pressure generated by rapid ohmic heating on the foil surface from the drive current. This expansion would exhibit a rapid expulsion of material best modeled by a short duration, larger amplitude acceleration compared to the interface accelerations measured later during the expansion phase. This acceleration may not have been completely observable early in time due to the rippled foil surface presumed to exist along the z-direction, and may have occurred over a short distance compared to the height of these ripples and to the observable foil thickness. The movement observed by the laser may have been the line-integrated average of this effect after the acceleration had already taken place, and therefore appeared in the longer timescale expansion plotted in Figure 6.16 as an initial velocity. A shot series with improved surface uniformity and backlighter images concentrated within this time window would be required to potentially resolve this initial acceleration, or to better constrain the delay time for the initial velocity model presented here. The magnitudes of the initial velocity coefficient ratios between the right and left sides are shown in Table 3. If these velocities were in fact caused by kinetic pressure expansion it was reasonable to expect that the expansion should have been the same on both sides of the foil since both sides received approximately the same energy deposition. The velocity ratios were close to unity for all the cases measured, particularly considering the uncertainties in the fit coefficients, although the existing uncertainties would have

also masked deviations from unity that could have indicated an unexpected asymmetry in the energy deposition between the right and left interfaces early in the drive.

Table 3 Ratios between the left side initial velocity and the right side initial velocity from empirical fit parameters for the bulk plasma expansion measurements.

Threshold	Large Offset	Small Offset
15%	1.05	1.17
Mean	1.00	1.09
85%	0.96	1.06

The acceleration coefficients from the fits represented the movement due to the net force created between the kinetic pressure inside the bulk plasma and the magnetic pressure surrounding it. These coefficients were quite sensitive to the delay time and in particular to the initial velocity coefficient, which was determined as an empirical fit rather than being included as a set value from detailed experimental measurements, so the acceleration values themselves, while still the best fit coefficients for the data, were intended for relative comparisons between cases and sides. The magnitude of the acceleration coefficient ratios between the right and left sides are shown in Table 4.

For the large offset case the left side displayed positive acceleration coefficients between $6.5 \text{ cm}/\mu\text{s}^2$ and $7.3 \text{ cm}/\mu\text{s}^2$, which were the largest acceleration coefficients measured and indicated a strong deceleration. This acceleration toward the midplane was exactly what was predicted from the slab acceleration model with an acceleration vector pointing into the bulk plasma. The values on the right side of the foil; however, were smaller than values on the left side by a factor of 3.5 for the 15% threshold, and the 85% threshold indicated a mild deceleration of $-0.37 \text{ cm}/\mu\text{s}^2$. Assuming the presence of the initial velocity was an accurate approximation, the detailed acceleration behavior for this

Table 4 Ratios between the left side acceleration and the right side acceleration from empirical fit parameters for the bulk plasma expansion measurements.

Threshold	Large Offset	Small Offset
15%	3.50	0.75
Mean	5.21	0.69
85%	19.7	0.75

interface was difficult to determine from the given data because the acceleration values were simply too sensitive to the initial expansion conditions.

For the small offset case, the left and right sides displayed qualitatively the same deceleration behavior, indicating that the bulk plasma expanded from both sides but was accelerated back toward the midplane. Quantitatively, the acceleration coefficients on the left side were 25% to 31% smaller than on the right side, but this apparent difference was well within the sensitivity of the acceleration coefficients to changes in the initial velocity. The most important observation from this case was that the initial expansion of the bulk plasma region created a geometry that no longer contained a thin foil nearly centered on the midplane, and that both interfaces were accelerated inward.

The expansion of the bulk plasma was also measured in total width, Δ , between the mean thresholds on either side of the foil. In addition, the center position of this bulk plasma was calculated and plotted as a function of time to examine any translation. These measurements are shown in Figure 6.17. The expansion velocity was 1.4 ± 0.2 cm/ μ s for the large offset case and was 1.4 ± 0.1 cm/ μ s for the small offset case, indicating that the offset had no appreciable effect on the overall expansion rate. The bulk plasma position plots contained zero detectable translation. The bulk plasma appeared to expand about its initial location and did not translate toward the midplane. The expansion clearly changed

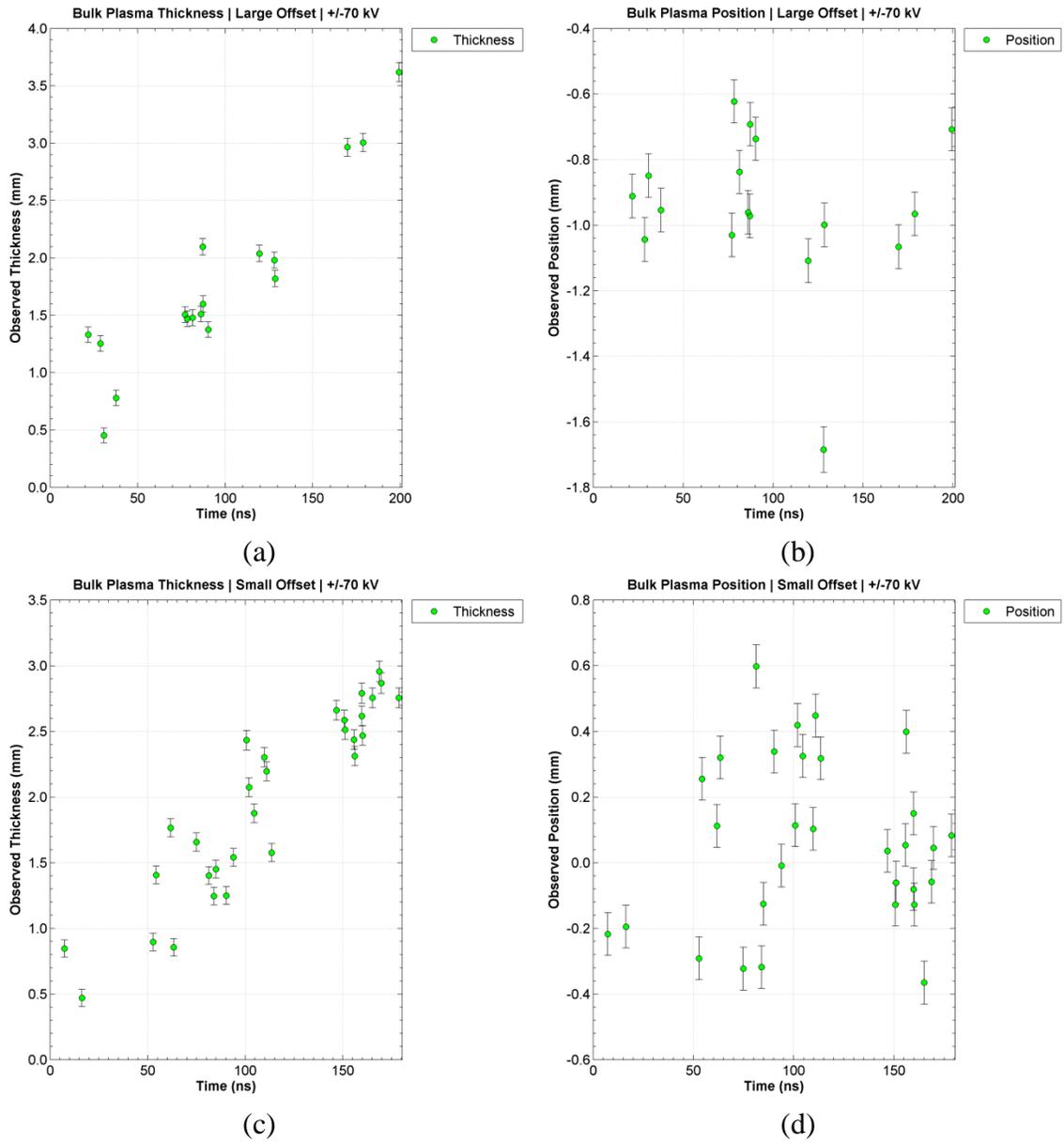


Figure 6.17 (a) Expansion measurements of the bulk plasma between the mean laser transmission thresholds for the large offset case. (b) Location of the bulk plasma center for the large offset case. (c) Expansion measurements of the bulk plasma between the mean laser transmission thresholds for the small offset case. (d) Location of the bulk plasma center for the small offset case.

the load dynamics from those predicted for a thin plasma slab from which acceleration toward the midplane would have been expected.

6.5 Magneto-Rayleigh-Taylor Instability Growth

The easily observable development of the MRT instability structure ($\lambda \sim 1$ mm) during the expansion phase allowed the instability growth to be measured from the data images as a function of time. This measurement was taken on both sides of the foil for both offset cases. These measurements were then compared to the expansionless slab model (ESM) growth predictions to investigate the effect of plasma expansion on the growth predictions. The planar geometry removed the difficulty of distinguishing the MRT instability from “sausage” and “kink” instabilities that do not occur for planar geometry, but would have been possible in a cylindrical geometry.

6.5.1 Experimental Measurements of MRT Growth

Tracking the growing amplitude of the complicated, multimoded instability structure required a figure of merit (FOM) to be chosen that could adequately approximate the amplitude of the entire instability. Ideally, the amplitudes of specific modes would have been tracked using spatial Fourier transforms of the parallel lineouts, but this option was not available due to inadequate resolution for wavelengths above 0.5 mm that dominated the mode structure for this instability. Consequently, the amplitude of the overall structure was used as a metric by measuring the distance between the 15% and 85% transmission thresholds. This FOM therefore tracked the overall expansion of mass within the ROIs. The progression of these amplitudes are plotted in Figure 6.18. Exponential models with an initial amplitude coefficient, A , and an exponential growth

rate coefficient, γ , were fit to the amplitudes. Therefore, γ represented an average growth rate over the entire drive current risetime.

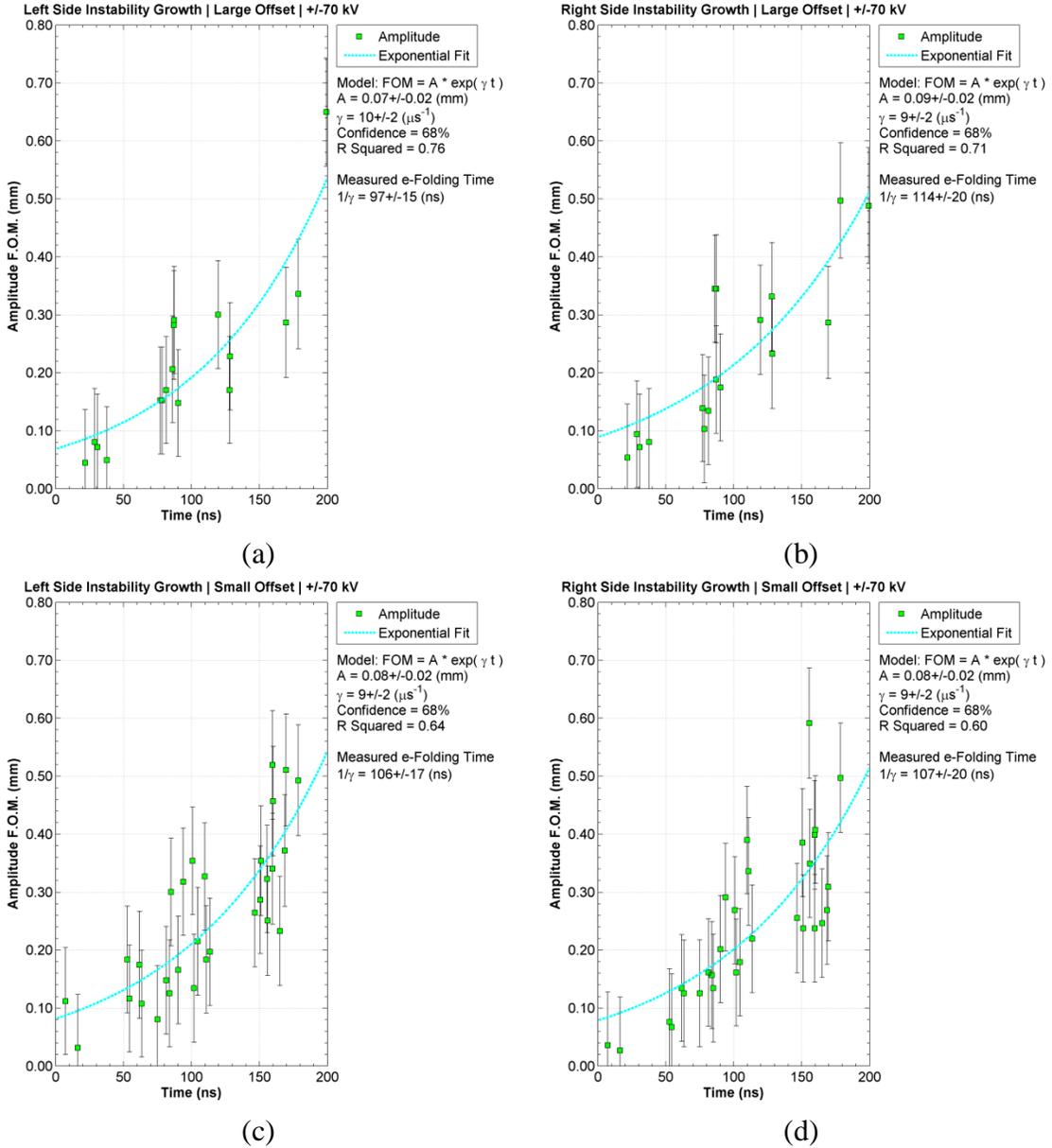


Figure 6.18 MRT growth rate figure of merit measurements for: (a) the left side of the large offset case, (b) the right side for the large offset case, (c) the left side for the small offset case, and (d) the right side for the small offset case. Empirical best fit curves are shown to measure the exponential growth rates for all cases.

The most striking observation from the FOM measurements, and from the derived γ values, was that the growth rates were indistinguishable from each other, within the uncertainty of the regression coefficients, for both sides of the foil and for both offset cases. The regression coefficients are summarized in Table 5 along with the e-folding times, $1/\gamma$.

Table 5 Initial amplitudes, growth rates, and corresponding e-folding times for a given foil side and offset.

Offset / Side	A (mm)	γ (μs^{-1})	$1/\gamma$ (ns)
Large / Left	0.07 ± 0.02	10 ± 2	97 ± 15
Large / Right	0.09 ± 0.02	9 ± 2	114 ± 20
Small / Left	0.08 ± 0.02	9 ± 2	106 ± 17
Small / Right	0.08 ± 0.02	9 ± 2	107 ± 20

The measured e-folding times were all roughly 100 ns, which was quite mild, given that the drive current rise time was approximately 170 ns. The indistinguishable growth rates between cases suggested that both foil interfaces experienced approximately the same average acceleration vector, with the direction pointing into the bulk plasma to drive the instability, and that this acceleration did not strongly depend on the initial foil offset.

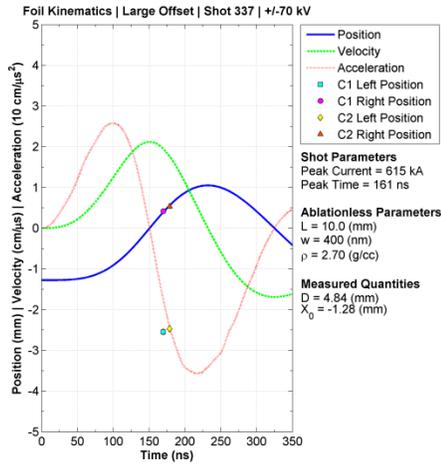
6.5.2 Comparison to Expansionless Slab Model Growth Predictions

The growth rates from the expanding foil plasma measured from the data images were compared to the predictions from the ESM. The foil acceleration was calculated using Equation 3.14 with the experimental drive current, anode spacing, and initial position as inputs to find what the load performance would have been according to the ESM. The acceleration was integrated using a 4th order Runge-Kutta method to calculate the foil velocity and position with zero for the initial velocity and the measured foil offset for the initial position. The ESM did not allow for expansion, so the initial density for Al,

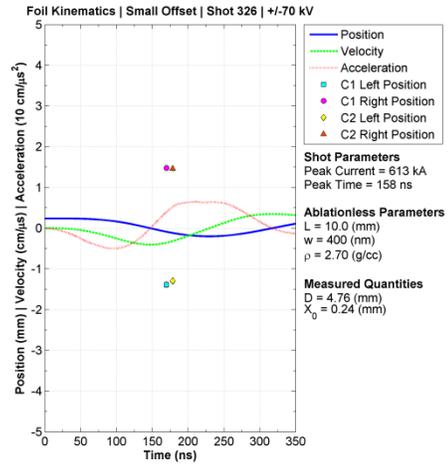
2.7 g/cm³, was used throughout the calculation to conserve mass. The magnetic fields on either side of the foil were calculated using the foil position at each time step according to Equation 3.10 for the left side, and Equation 3.11 for the right side. The instantaneous growth rates, γ , for the instability were also calculated using the instantaneous acceleration and magnetic fields. The Rayleigh growth rate scaling given by Equation 3.111, as well as the generalized growth rate for this 3-region geometry given by Equation 3.144, along with the integrated number of e-foldings, Γ , for each model and shown in Figure 6.19. The instability wavelength in the y-direction was set to be 1 mm, which was in the middle of the 0.8 mm and 1.2 mm modes measured from the parallel lineouts. Since no measurements existed to determine the z-direction wavelength, the only reasonable assumptions for this wavelength were to neglect all effects from the z-direction using a zero wavenumber, or to assume that the z-direction wavelength would develop in a similar manner to the y-direction wavelength and simply set them equal to one another. The latter option was chosen for this illustrative analysis to allow the magnetic field to create a stabilizing effect for the generalized model to contrast with the Rayleigh model. The absolute value for the Rayleigh scaling model was used rather than just the imaginary component because the 400 nm thick foil was assumed to develop nearly complete instability feed-through, so the absolute value included the total MRT growth that would develop on either side of the foil while accelerating on either side of the midplane. The feed-through factor with a 400 nm thick foil for the Rayleigh-Taylor scaling, $\exp(-k \Delta)$ as shown in Equation 3.121, was 0.998 for a 1 mm wavelength, 0.975 for a 100 μm wavelength, and 0.778 for a 10 μm wavelength. This indicated that even though the right interface of the foil was stable according to the acceleration vector and

Atwood number, the feed-through effect was expected to generate instability growth on the right interface nearly equal in magnitude to the unstable left interface for mode wavelengths around 1 mm. These predictions are plotted in Figure 6.19. Without the absolute value assumption, the imaginary component for the Rayleigh growth model would have produced instability on the left interface, forming the first peak in Figure 6.19 (e) and (f), and the real component, which was equivalent to reversing the sign for the interface Atwood number, would have produced the second peak on the right interface as the foil oscillated back toward the midplane. The large offset case and small offset case examples are plotted side by side to highlight the differences in the predictions. The Rayleigh curves did not include stabilization due to magnetic tension. The actual measured locations for the mean thresholds around the bulk plasma region are plotted with the predicted foil positions to show the very different behavior caused by expansion.

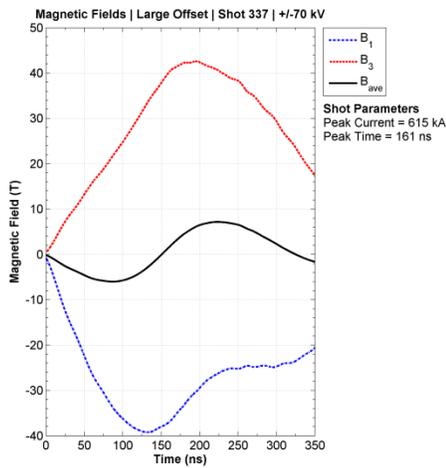
The ESM predicted that the imbalance in magnetic pressures acting on each side of the foil would create a net restoring force that would accelerate the entire foil mass toward the midplane, causing the foil to displace and oscillate between the anodes. These displacement predictions are shown in Figure 6.19 (a) and (b), where the initial offset dependence of the oscillation magnitude about the midplane is clearly apparent. The foil was assumed incompressible by assuming that the entire foil would experience the same acceleration due to this net force. The actual positions of the mean thresholds plotted for the example shots in Figure 6.19 (a) and (b), as well as the entire data image sets for both offset cases shown in Figure 6.4 and Figure 6.5 for the large offset case, and in Figure 6.10 and Figure 6.11 for the small offset case, illustrated how the foil did not move as a single mass, but instead expanded about its initial location. This expansion was



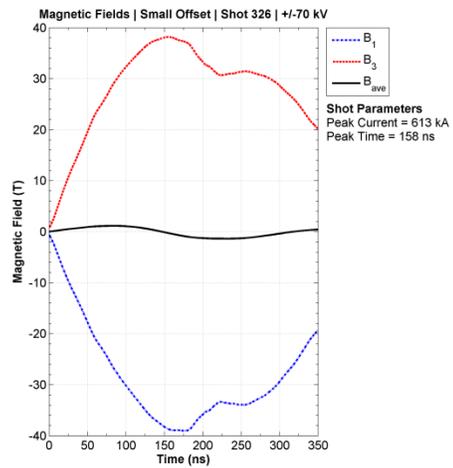
(a)



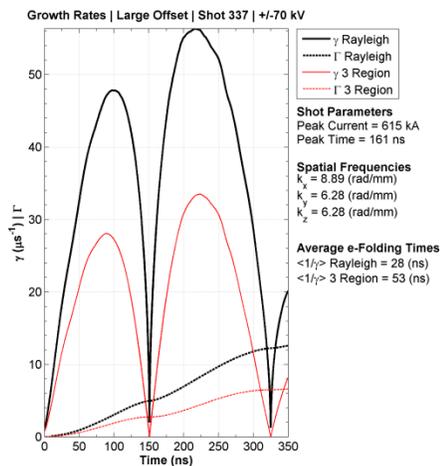
(b)



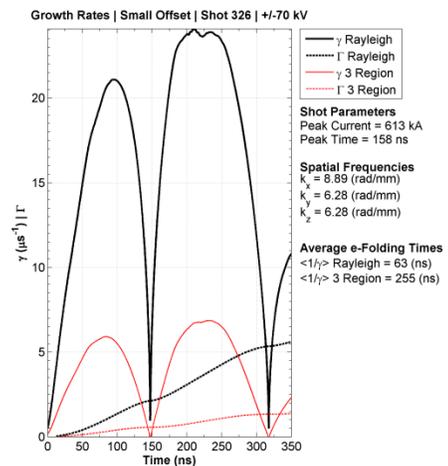
(c)



(d)



(e)



(f)

Figure 6.19 The predicted foil kinematics according to the ESM, along with actual measured initial positions, for (a) the large offset case, and (b) the small offset case. The resulting magnetic fields from the drive current and load inductance for (c) the large offset case, and (d) the small offset case. The predicted instantaneous growth rates and integrated e-foldings according to the Rayleigh model and the generalized model for (e) the large offset case, and (f) the small offset case.

summarized as a function of time and was shown to occur about the initial foil position, rather than experience any appreciable displacement, in Figure 6.17.

The calculated magnetic fields from the ESM had similar qualitative shapes for both offset cases, although the quantitative nature for each case varied due to the difference in inductance from the different foil position oscillation magnitudes. The offset foil side closest to the anode developed a faster rising magnetic field due to the initially lower inductance for this side with a partial double peak feature created from the foil oscillation. The opposite side developed a slower rising, but ultimately larger singly peaked magnetic field where the single maximum field magnitude occurred when the foil position reached its maximum displacement toward the opposite anode. The peak magnetic fields were approximately 40 T (0.40 MGauss), although the oscillation features varied in magnitude depending on the initial foil position. The average net magnetic field acting on the entire foil mass was approximated by assuming that the foil was very thin compared to the offset from the midplane, and that this single net magnetic field produced the restoring force.

The predictions shown in Figure 6.19 (e) for the large offset case, and in (f) for the small offset case, highlighted how the ESM acceleration, and resulting MRT growth predictions for both the Rayleigh and general models, reduced as the initial foil offset approached the midplane. The predicted peak instantaneous growth rates from the Rayleigh model for the large offset case were 5.6 times larger than the measured values, and the time averaged growth rates were 3.5 times larger than the measurements. The predicted peak growth rate for the generalized 3-region model, again assuming a z-

direction wavelength equal to the y-direction wavelength, were 3.3 times larger, and 2 times larger for the time averaged growth rates.

6.5.3 The Role of Plasma Expansion on Instability Growth

The deviation from the ESM geometry created by the expanding bulk plasma was the most likely reason that the measured MRT growth rate on the left side of the foil for the large offset case was smaller than predicted from the ESM, that the MRT instability grew on the right side of the foil for the large offset cases, and the MRT instability developed on both sides of the foil for the small offset case. This expansion was large enough to completely decouple the interfaces from each other by removing the feed-through effect since, for even a 1 mm expansion and the 1 mm instability wavelength, $k\Delta$ was 6.28, so $\exp(-6.28) = 0.002$. Additionally, the magnetic fields from each side of the foil exactly canceled between the current carrying foil interfaces, which removed any internal magnetic pressure and left only kinetic pressure. This internal kinetic pressure attempted to balance against the external magnetic pressure. At early times, when the drive current was low and the resulting magnetic pressures were also low, the kinetic pressure generated a net outward acceleration, expanding the bulk plasma. As the current and magnetic pressures increased, the interfaces experienced a net inward acceleration that drove the MRT growth measured in Section 6.5.1. This initial acceleration and then deceleration occurred instead of the complete displacement predicted by the ESM. The net accelerations resulting from the internal kinetic and external magnetic pressures were what drove the MRT instability with similar growth rates measured for all sides and offset cases. Since the current, and resulting magnetic pressure, on the side containing the initial offset was larger than the opposite side, the net acceleration on the offset side

would have been expected to be greater compared to a centered case, and the growth rate on the opposite side reduced in the same manner compared to a centered case. The measured growth rate on the left side for the offset case was $10 \pm 2 \mu\text{s}^{-1}$ compared to $9 \pm 2 \mu\text{s}^{-1}$ for the right side. These measurements could not be distinguished from each other, perhaps a result of the relatively modest 22% average offset for this case. A larger offset may have produced a larger difference between growth rates that could have been distinguished, but with reduced observation time because of anode plasma closure.

The measured net acceleration for the large offset case on the left side was $1.6 \pm 0.6 \text{ cm}/\mu\text{s}^2$, and $1.3 \pm 0.6 \text{ cm}/\mu\text{s}^2$ for the large offset case on the right side, and for both sides of the small offset case, according to the Rayleigh growth rate scaling using the measured k_y wavenumber as the total wavenumber, k . If k_y was assumed equal to k_z , the accelerations were reduced by $\sqrt{2}$. These measurements, along with the best fit acceleration coefficients from Figure 6.16, are shown in Table 6.

Table 6 A comparison between the acceleration coefficients and the acceleration required to generate the measured MRT growth from the Rayleigh scaling.

Location	Large Offset Left Side ($\text{cm}/\mu\text{s}^2$)	Large Offset Right Side ($\text{cm}/\mu\text{s}^2$)	Small Offset Left Side ($\text{cm}/\mu\text{s}^2$)	Small Offset Right Side ($\text{cm}/\mu\text{s}^2$)
15%	6.6 ± 3.5	1.9 ± 2.5	2.1 ± 3.0	-2.8 ± 2.3
Mean	6.5 ± 3.4	1.3 ± 2.3	3.7 ± 3.0	-5.3 ± 2.3
85%	7.3 ± 3.6	-0.4 ± 2.4	4.3 ± 3.2	-5.8 ± 2.8
$\gamma^2 / k_y (k_y=k)$	1.6 ± 0.6	-1.3 ± 0.6	1.3 ± 0.6	-1.3 ± 0.6
$\gamma^2 / k (k_y=k_z)$	1.1 ± 0.4	-0.9 ± 0.4	0.9 ± 0.4	-0.9 ± 0.4

Direct comparison between the measurements and the acceleration coefficients was difficult due to the sensitivity of the coefficients to the initial velocity coefficients; however, the accelerations were the correct order of magnitude, and in the correct direction, except on the right side for the large offset case. The right side for the offset

case may have required a better knowledge of the initial velocity, and perhaps of the interface positions, later in time since the uncertainties for this case were large enough to change the direction of the acceleration vector. The observation that the ablation streamers appeared longer on this side than on the left side may have also contributed to the apparent lack of measured deceleration at this interface if the ablation streamer material, which became more chaotic in structure with streamers bending and merging together, increased the opacity to the backlighter farther away from the initial position than would have been measured from a centered foil.

The fact that the measured acceleration coefficients were larger than the acceleration required to produce the measured growth rate may have indicated that damping of the instability was occurring due to the magnetic field, as predicted by the generalized growth rate. This growth rate, in the single interface limit where $\coth(k\Delta) \rightarrow 1$, and $V_{\text{au}} \rightarrow 0$, was given by Equation 3.151. The magnitude of this damping could have been quantified in principle, but the uncertainties in the acceleration coefficients were too large to produce a detailed result, and a detailed z-direction wavelength would also have been required beyond the simplistic $k_y=k_z$ assumption, which was not possible with the hardware utilized for these experiments. The MRT growth remained robust even with stabilization.

The measured acceleration coefficients, growth rates, and physical interpretation resulted in a framework for describing the MRT instability growth observed in the laser backlighter images, perhaps seeded by the ablation streamer structure that characterized the initiation phase. Beyond the instability growth rate measurements themselves, the presented work has provided valuable insight regarding which aspects of the

experimental configuration and theoretical models (plasma expansion) must be improved for future measurements and analysis of this phenomenon.

CHAPTER 7

SUMMARY AND SUGGESTIONS FOR FUTURE WORK

The ablation dynamics of wire array cores in Z-pinch loads was investigated using X-pinch as X-ray sources for point-projection radiography. These ablation dynamics were of interest because nonuniformities in the ablation structure were believed to be a contributing seed for the magneto-Rayleigh-Taylor (MRT) instability that occurred on the outer edge of the wire array plasma as it implodes onto the central axis, affecting the X-ray performance from the Z-pinch loads. Correlations between these nonuniformities may also be a contributing factor for the drop in X-ray yields when the number of wires is increased beyond a certain threshold. These wire cores were observed to ablate in an azimuthally-correlated manner between wires if the wire spacing was sufficiently close (240 μm), where gaps in one wire core became correlated with gaps in a neighboring wire core. The correlation was quantified and a transition region between correlated and uncorrelated ablation dynamics was identified between 240 μm and 2.47 mm. Two possible physical mechanisms for the correlation were identified.

The first possible mechanism was current shunting through azimuthally-oriented current paths through neighboring wire coronal plasmas. When a break in one wire generated an imploding magnetic bubble, these current paths shunted current away from the more inductive bubble front through a neighboring wire corona. The current then shunted back to the original wire to maintain inductively-favorable parallel current paths.

The enhanced current in the neighboring wire at the same axial location as the imploding bubble enhanced the ablation rate from Joule heating and led to azimuthally-correlated wire core breaks.

The second possible mechanism was the magnetic field distortion near a wire core break. When the ablating wire core at a fixed axial location ran out of material, the coronal plasma at that location was swept towards the axis, creating an imploding magnetic bubble. The distorted magnetic field due to this bubble increased the magnetic pressure acting on the coronal plasma around a neighboring wire core at that same axial location, which resulted in an increased ablation rate from Joule heating and azimuthally-correlated wire core breaks.

Future investigations of azimuthally-correlated ablation should consider characterizing the transition region between 240 μm and 2.47 mm by adjusting the wire spacing over this range. Additionally, the coupling between more than 2 wires would be of interest. This could be accomplished by initially using three wires and then increasing the wire number to measure how the ablation correlates from one to the next. The wire spacing for these experiments should also be swept to see if the transition region is the same when more than two wires are present.

The magneto-Rayleigh-Taylor instability of a 3-region planar geometry was investigated analytically using ideal magnetohydrodynamics. This treatment allowed for arbitrary densities and magnetic fields parallel to the interfaces, and to each other, with arbitrary amplitudes. The force driving the instability was an arbitrary combination of magnetic pressure and kinetic pressure. The dispersion relation, growth rates, stability

conditions, and feed-through factors for the instability coupling between the interfaces were analytically determined. An anisotropy in the instability evolution was determined where instability modes with wavenumbers parallel to the magnetic field experienced a stabilizing effect from the magnetic tension, whereas instability modes with wavenumbers perpendicular to the magnetic field were unaffected. Additionally, an expansionless slab model (ESM) was developed and used to predict load kinematics for a thin foil placed between current return plates and pulsed with an arbitrary drive current in a planar geometry.

A circuit model, based on LTSPICE, was constructed to simulate the pulsed power capabilities of the MAIZE facility's linear transformer driver (LTD) module at the University of Michigan's Plasma, Pulsed Power, and Microwave Laboratory. This circuit model contained a modular block architecture and nonlinear ferromagnetic core inductors. The model accurately reproduced the experimentally determined LTD module performance. The drive currents from this model were used as inputs for the ESM, and the resulting accelerations were used to calculate predicted MRT growth rates. The peak predicted e-folding times were 13.3 ns or less.

The MAIZE facility was constructed to experimentally measure the MRT growth rates for 400 nm thick aluminum foils in the double return plate geometry. The LTD, load hardware, support systems, and laser backlighter diagnostic were assembled for this experiment. Two cases were tested where the foil's initial position relative to the midplane between the return plates was varied: a large offset versus a small offset. The small offset foil was placed as close to the midplane as physically possible. The large offset case was predicted to generate a large acceleration of the foil toward the midplane

to drive MRT instability growth, whereas the small offset case was predicted to generate a nearly zero acceleration and consequently to exhibit much lower MRT growth. The experimental results; however, revealed a foil plasma expansion dynamic that was not accounted for in the ESM predictions. This significant plasma expansion served to invalidate the ESM, to decouple the foil interfaces, and to create a geometry where the large external magnetic pressure accelerated both interfaces inward against the internal kinetic pressure. The measured MRT e-folding times were all approximately 100 ns for both foil interfaces for either offset case. The dominant instability wavelength was approximately 1 mm. For the large offset case, the interface initially away from the midplane was unstable as expected from the measured interface acceleration, but the interface facing the midplane exhibited the same instability growth, which was unexpected. The acceleration behavior for this interface could not be definitively measured to justify the observed MRT growth and remains a subject of future research. For the small offset case, this expansion made both foil interfaces unstable with consistent acceleration measurements.

Future investigations of MRT growth may wish to consider focusing on improving the laser diagnostic. Some particularly useful improvements would be to increase the laser energy entering the frequency doubling crystal by expanding the beam size on the crystal and removing neutral density. This would improve image contrast and facilitate adding additional beam lines. The most important beam line alterations would be to field an interferometer to measure the outer foil plasma's electron density, and to alter the load hardware to allow a laser to be passed perpendicular to the existing beam along the foil. This would allow the instability wavenumber parallel to the magnetic field

lines to be measured. These parameters are critical for applying the generalized 3-region instability growth scaling to the experimental measurements, and for investigating the anisotropy present in MRT due to the magnetic field, which is a key distinction between RT and MRT. The preliminary visible spectroscopic diagnostic should be improved to allow the plasma temperature in the lower density coronal plasma to be accurately measured. X-ray spectroscopy may be required to make temperature measurements of the more interesting dense plasma close to the foil's initial location. X-pinch backlighters would improve the image resolution and would allow the dense interior plasma to be imaged. Calibrated step wedges on the X-ray films would allow density measurement to be made of this higher density core region that would not be accessible with laser backlighting.

The foil load itself could be significantly improved. A new load cartridge with less variability in the installation location would reduce the variability in the initial observed foil thickness on the laser preshot images. The instability could also be seeded at the 1 mm observed wavelength by patterning the foil surface using laser ablation, or by patterning the return current plates to perturb the magnetic field. The initial surface condition for the tensioned foils used in the presented experiments was nearly impossible to determine because the foil had to move after being placed inside the experimental chamber and sealed under vacuum. The initial surface condition could be better characterized by depositing the metallic layer onto a low density foam or similar backing. This backing could be patterned prior to deposition to produce seeding perturbations, and the initial surface conditions could be well known since the backing would not require tensioning. The effect of foam on stabilization of MRT is another crucial area of

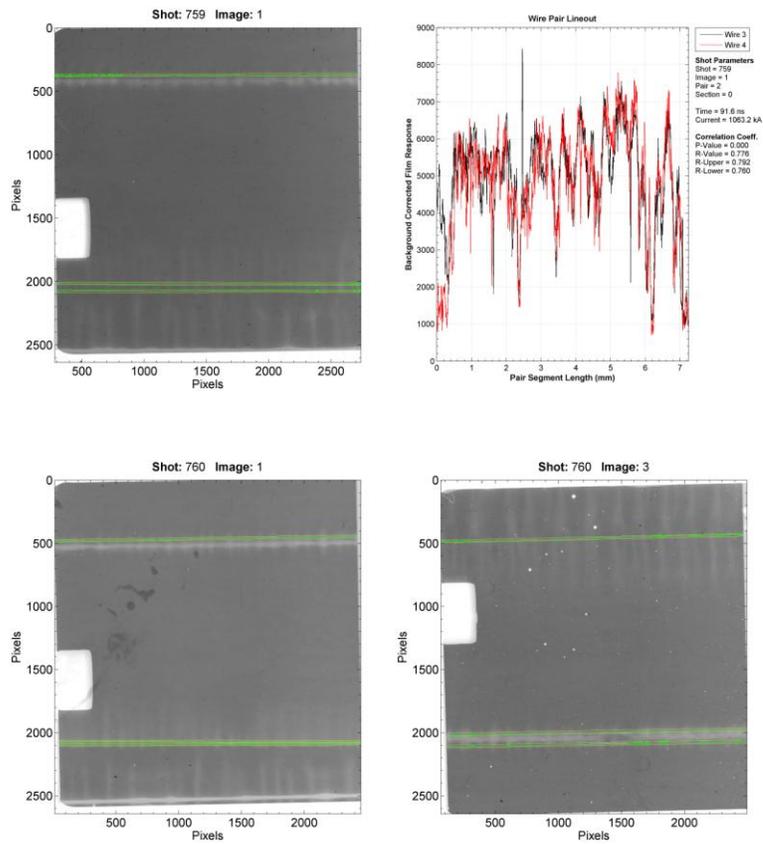
investigation. Finally, the equation of motion component of the MRT theory needs to be adapted to account for expansion of the ablation plasma.

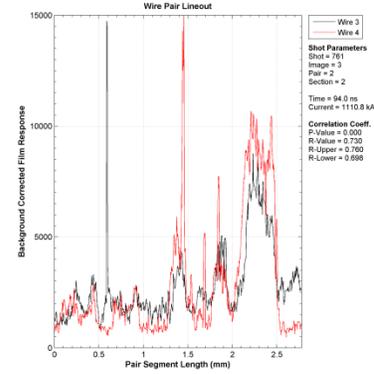
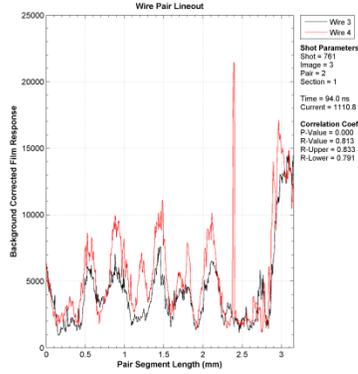
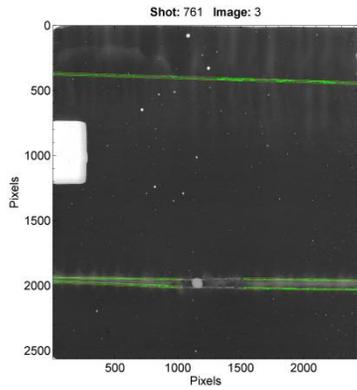
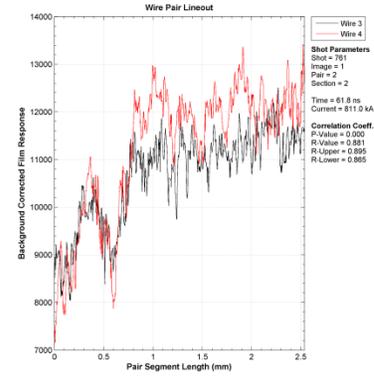
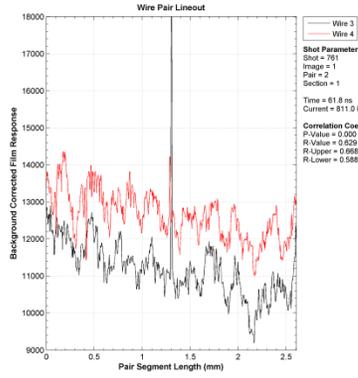
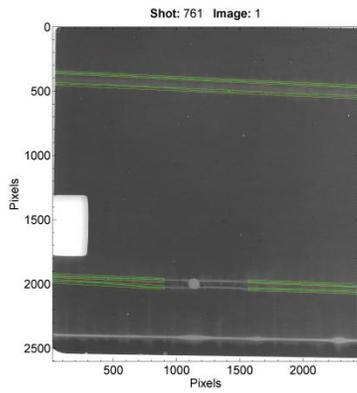
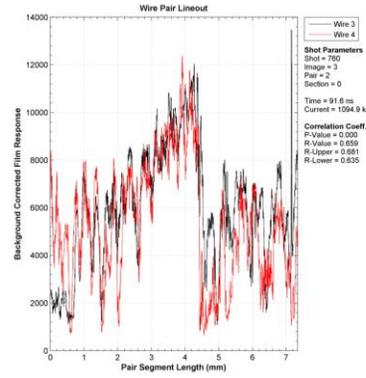
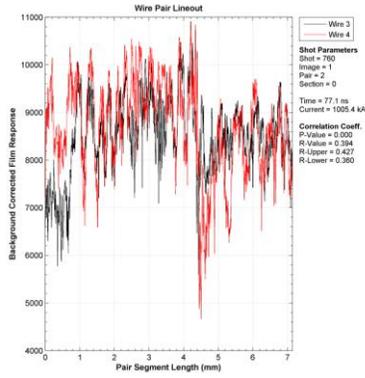
APPENDICES

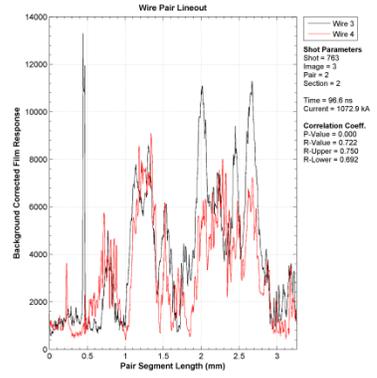
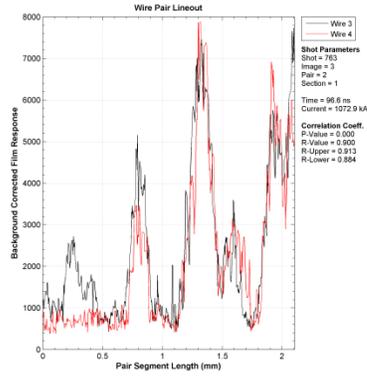
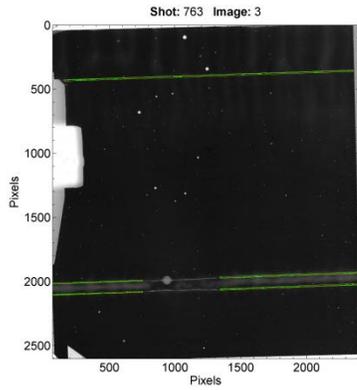
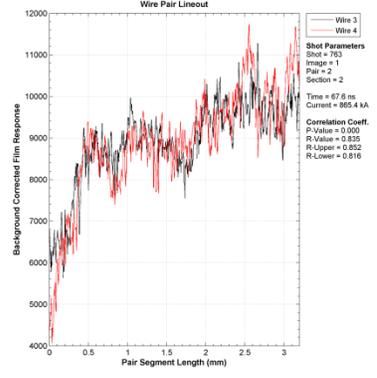
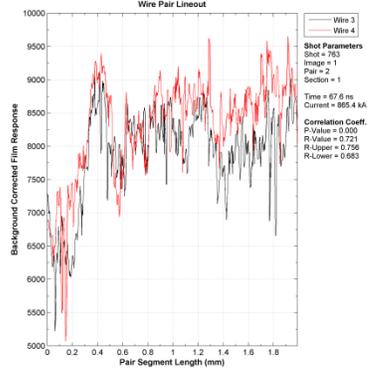
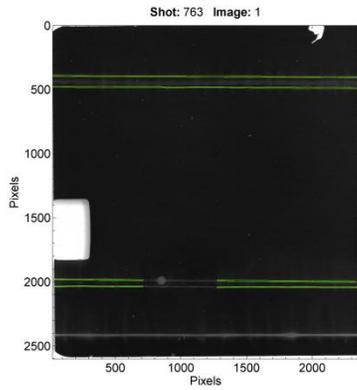
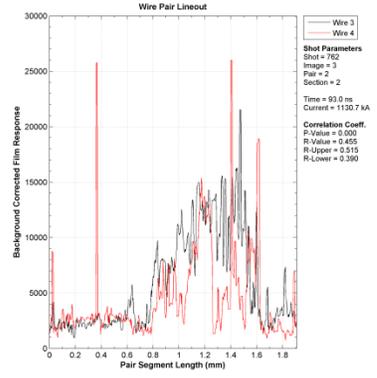
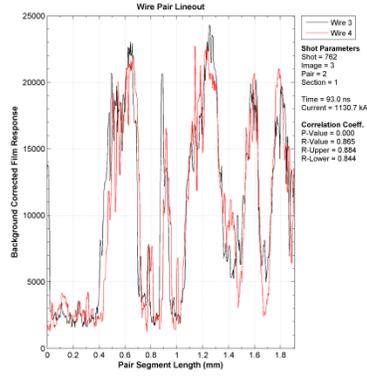
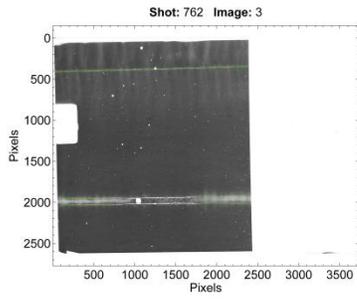
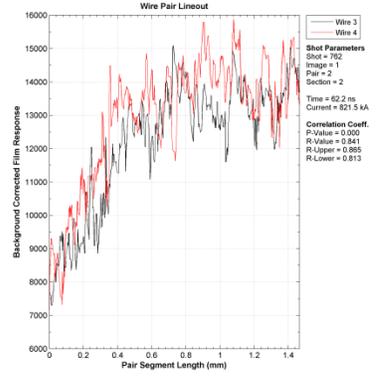
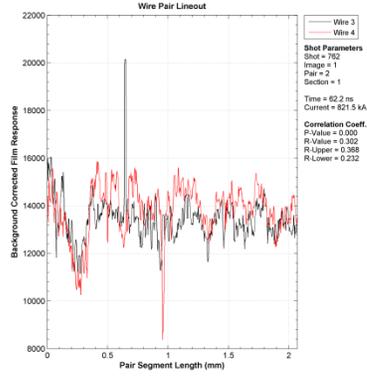
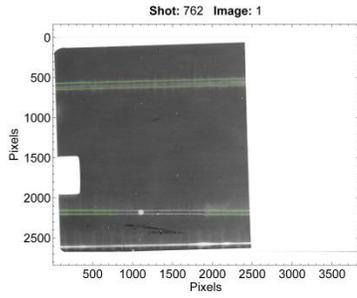
APPENDIX A

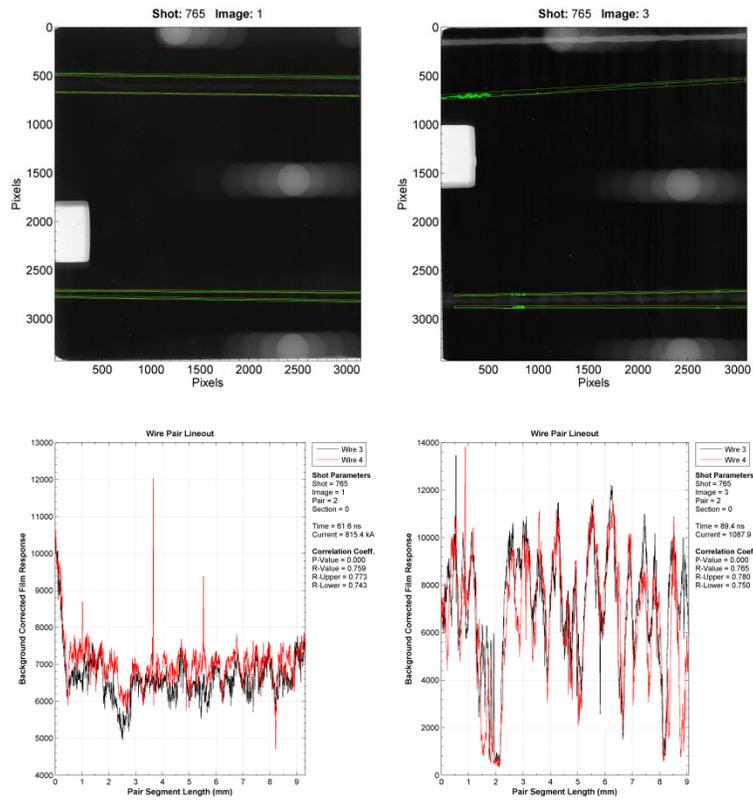
EXPERIMENTAL DATA

A.1 Azimuthally Correlated Ablation – Small Interwire Gap

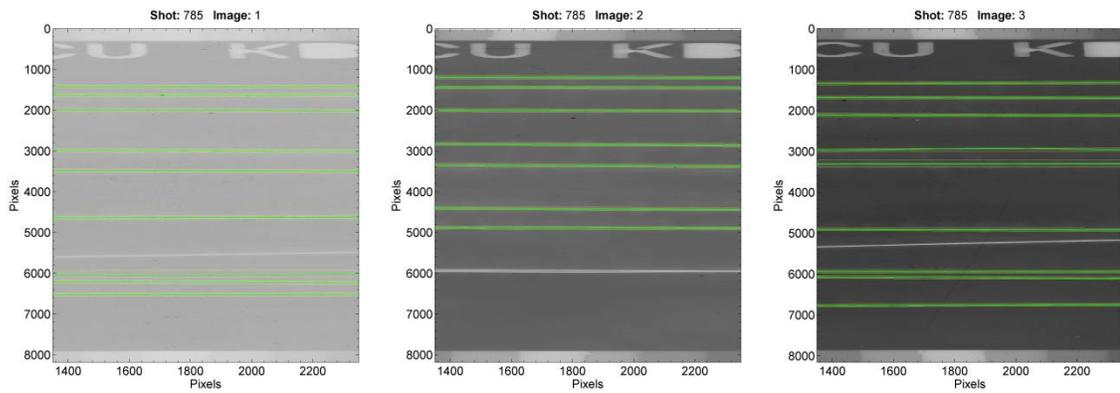


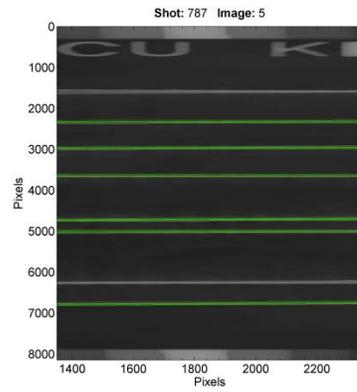
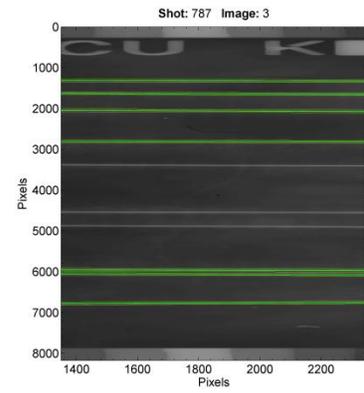
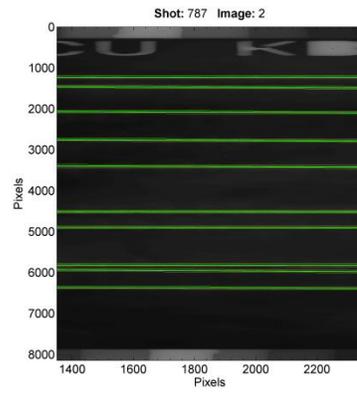
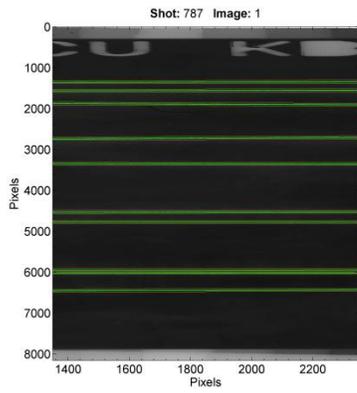
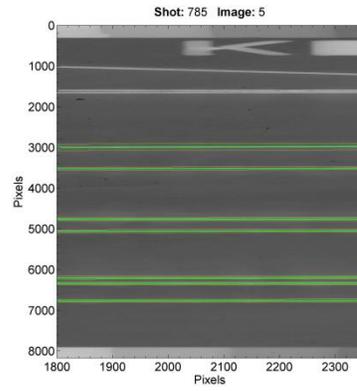
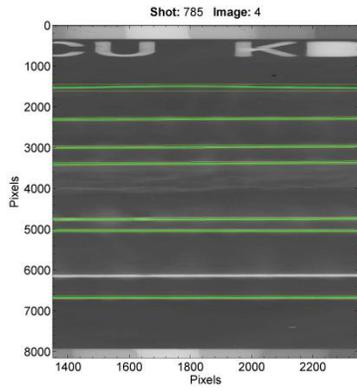


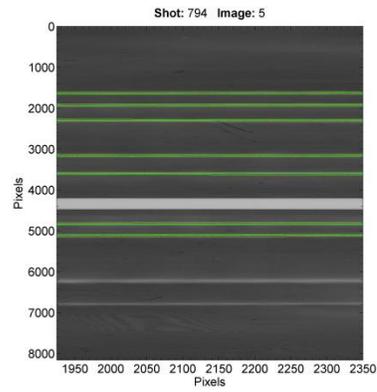
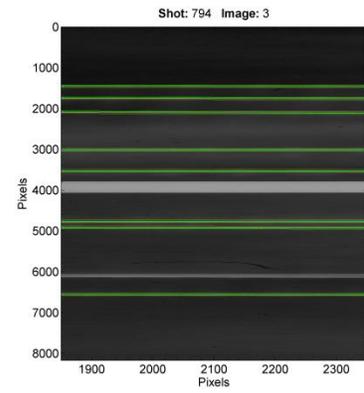
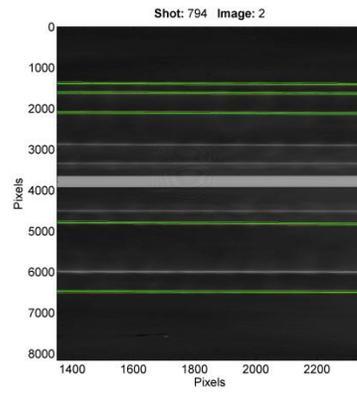
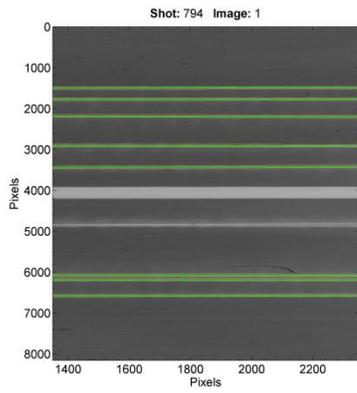
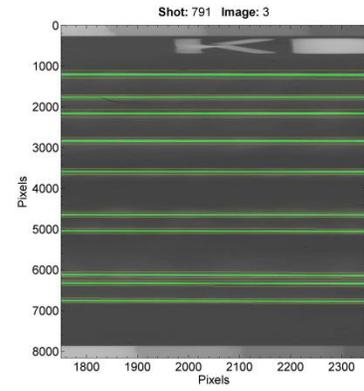
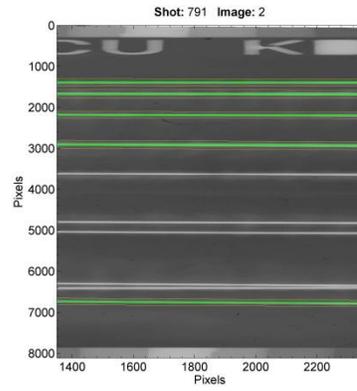
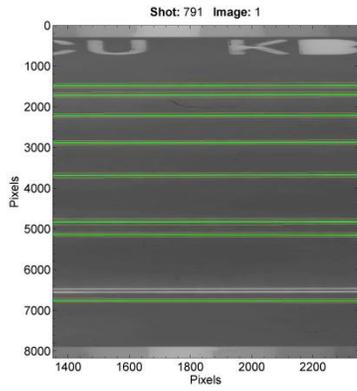


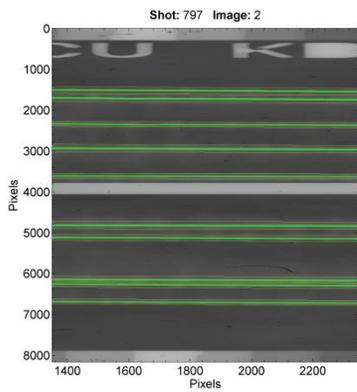
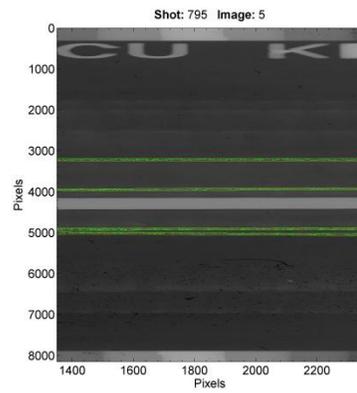
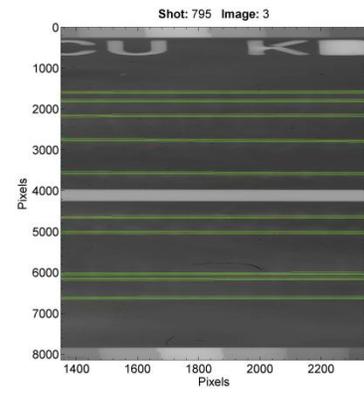
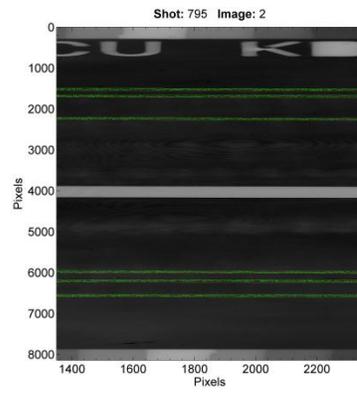
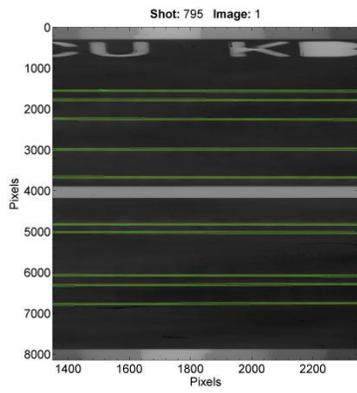


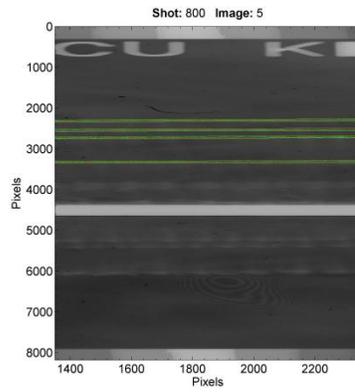
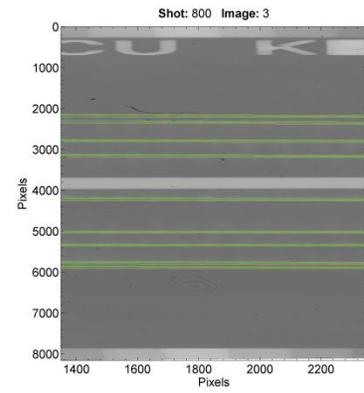
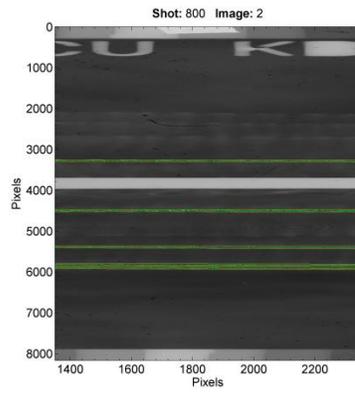
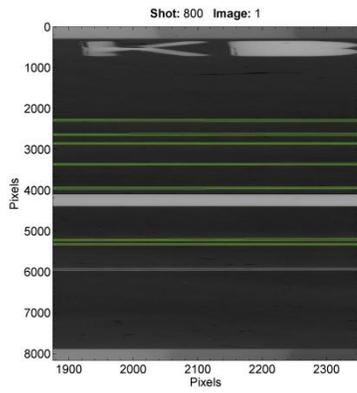
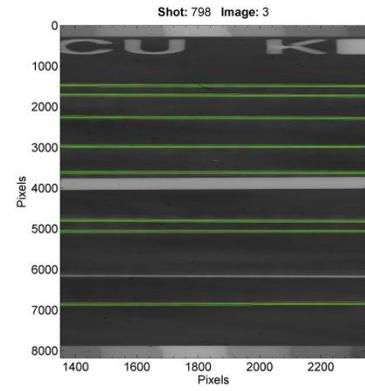
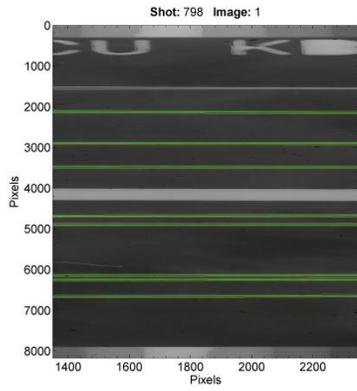
A.2 Azimuthally Correlated Ablation – Large Interwire Gap

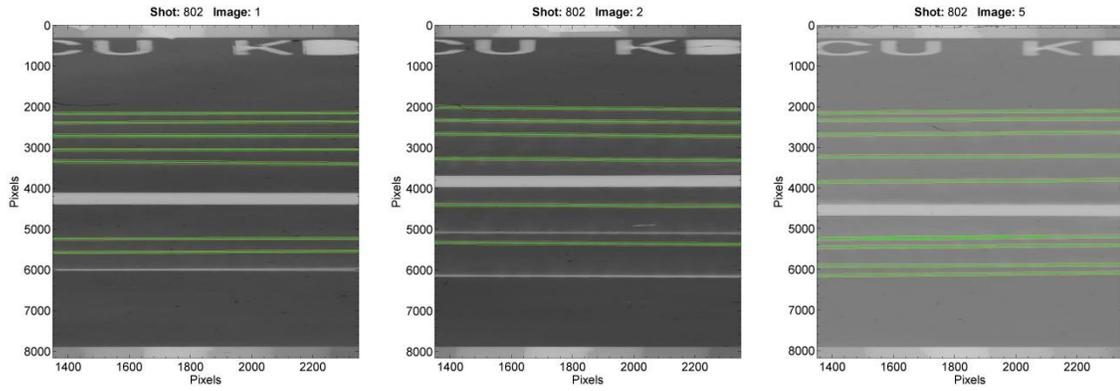




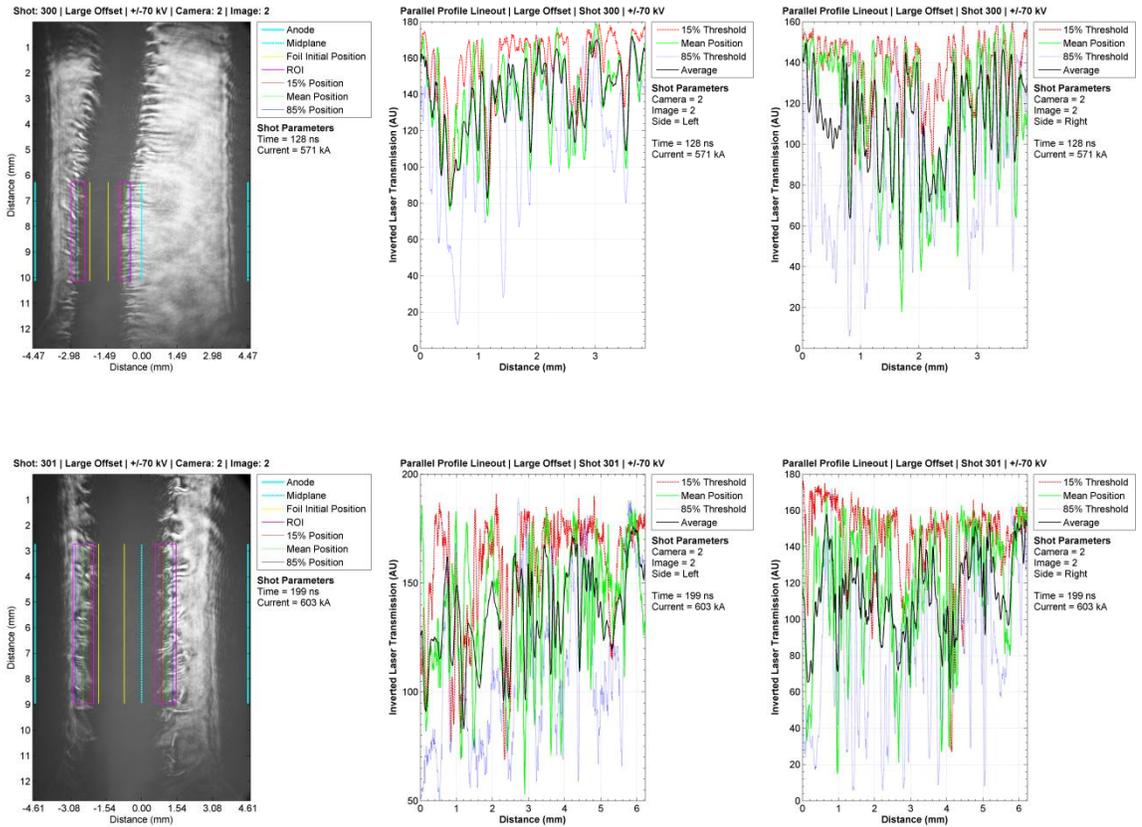




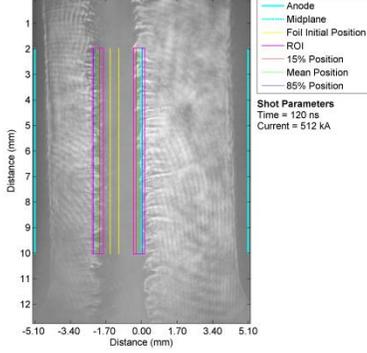




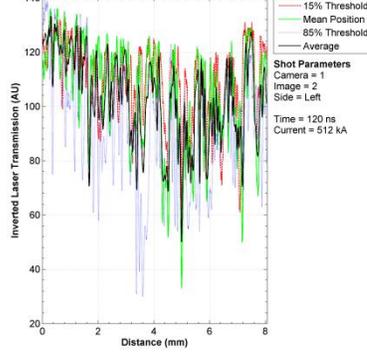
A.3 MRT – Large Foil Offset



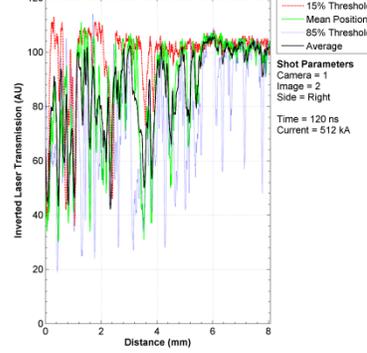
Shot: 306 | Large Offset | +/-70 kV | Camera: 1 | Image: 2



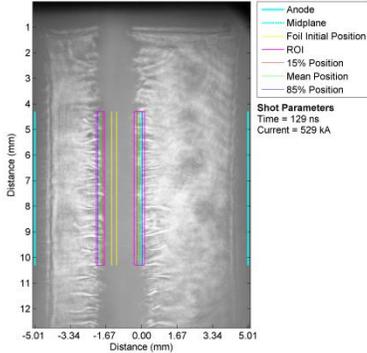
Parallel Profile Lineout | Large Offset | Shot 306 | +/-70 kV



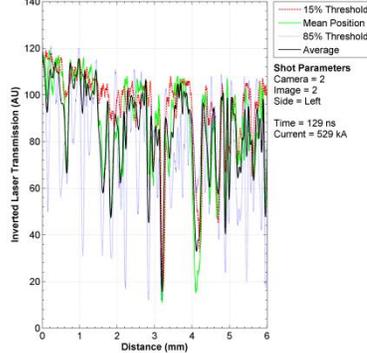
Parallel Profile Lineout | Large Offset | Shot 306 | +/-70 kV



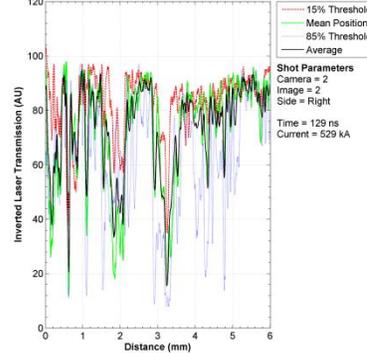
Shot: 306 | Large Offset | +/-70 kV | Camera: 2 | Image: 2



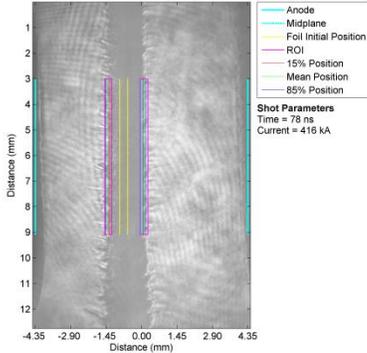
Parallel Profile Lineout | Large Offset | Shot 306 | +/-70 kV



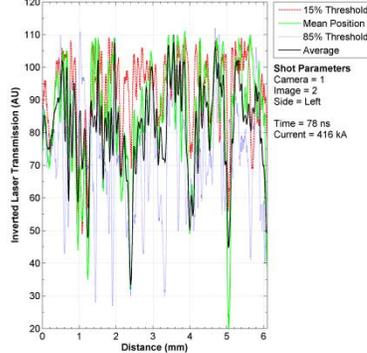
Parallel Profile Lineout | Large Offset | Shot 306 | +/-70 kV



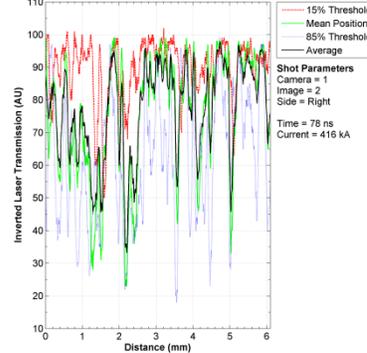
Shot: 308 | Large Offset | +/-70 kV | Camera: 1 | Image: 2



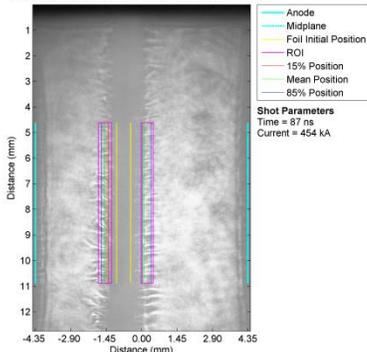
Parallel Profile Lineout | Large Offset | Shot 308 | +/-70 kV



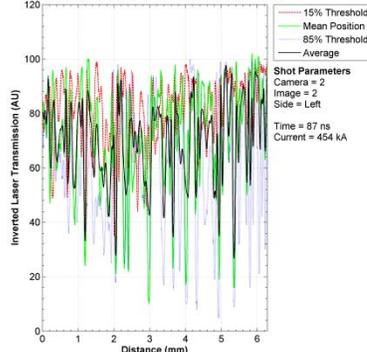
Parallel Profile Lineout | Large Offset | Shot 308 | +/-70 kV



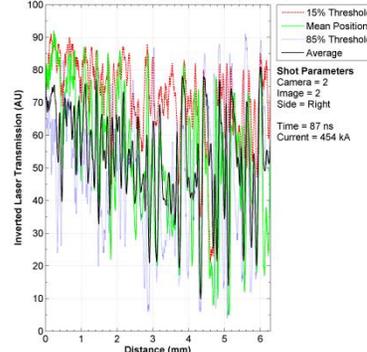
Shot: 308 | Large Offset | +/-70 kV | Camera: 2 | Image: 2

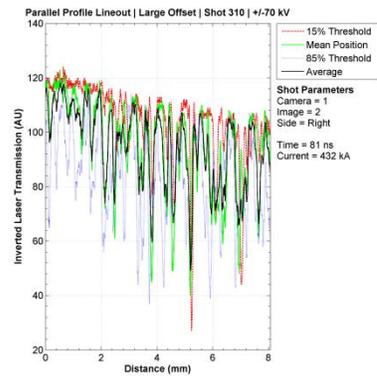
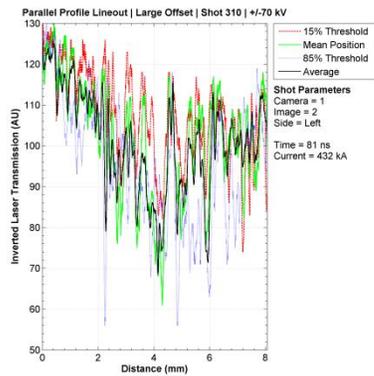
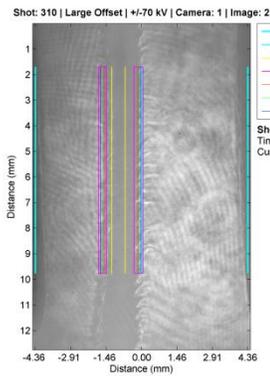
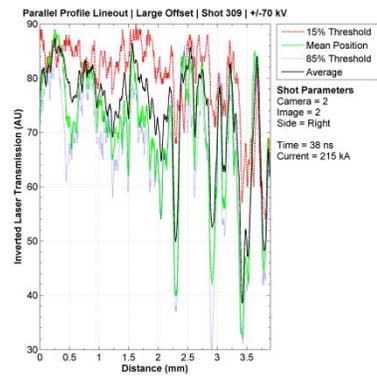
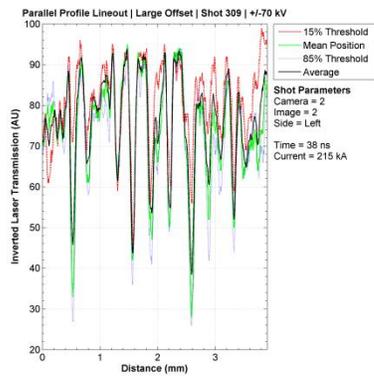
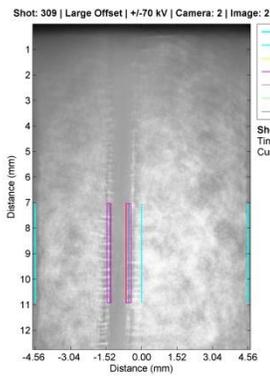
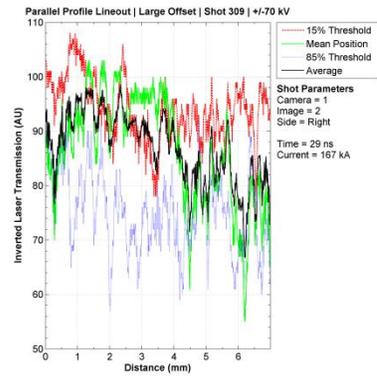
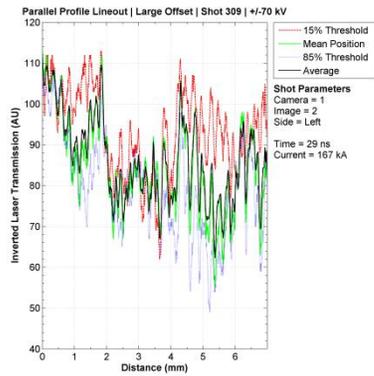
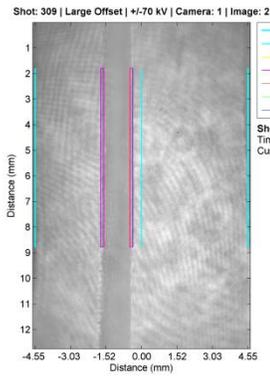


Parallel Profile Lineout | Large Offset | Shot 308 | +/-70 kV

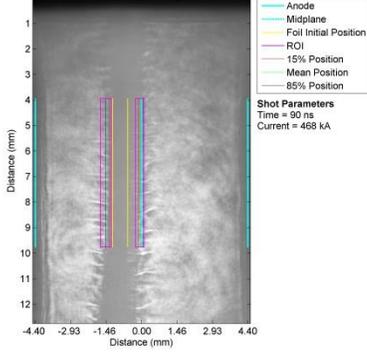


Parallel Profile Lineout | Large Offset | Shot 308 | +/-70 kV

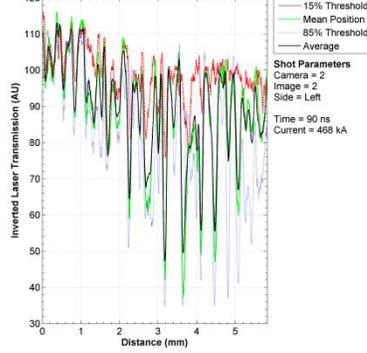




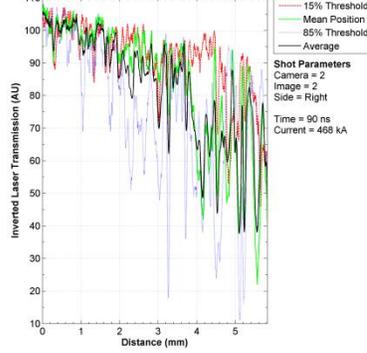
Shot: 310 | Large Offset | +/-70 kV | Camera: 2 | Image: 2



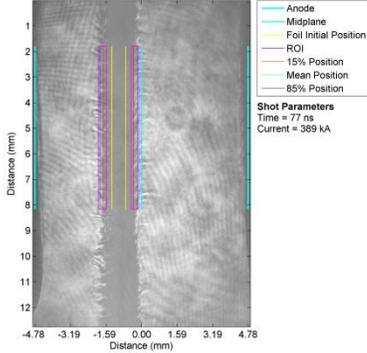
Parallel Profile Lineout | Large Offset | Shot 310 | +/-70 kV



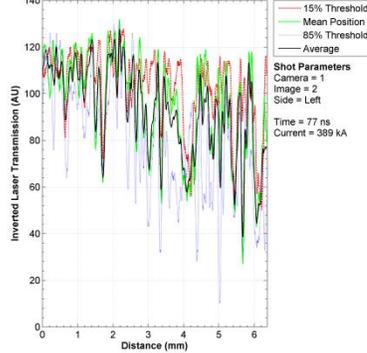
Parallel Profile Lineout | Large Offset | Shot 310 | +/-70 kV



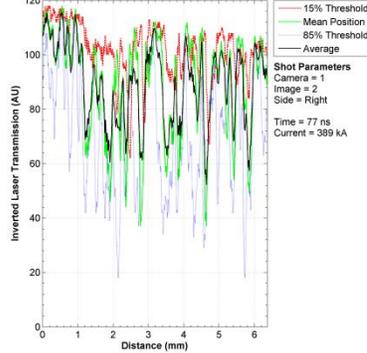
Shot: 311 | Large Offset | +/-70 kV | Camera: 1 | Image: 2



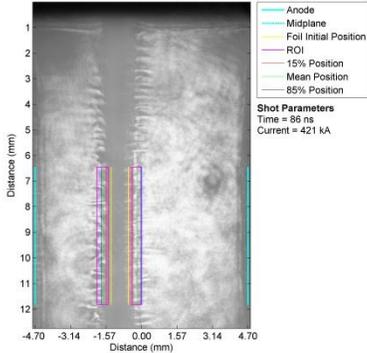
Parallel Profile Lineout | Large Offset | Shot 311 | +/-70 kV



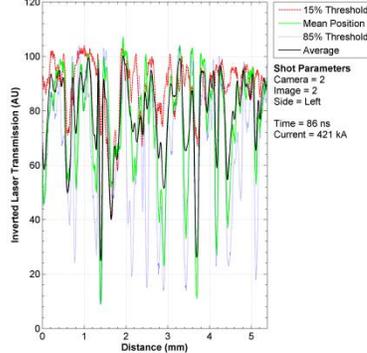
Parallel Profile Lineout | Large Offset | Shot 311 | +/-70 kV



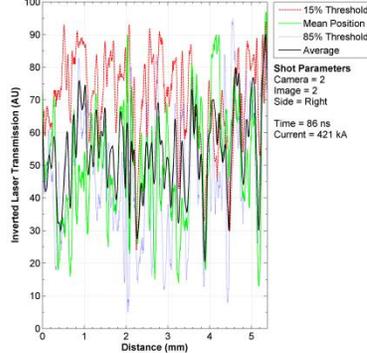
Shot: 311 | Large Offset | +/-70 kV | Camera: 2 | Image: 2



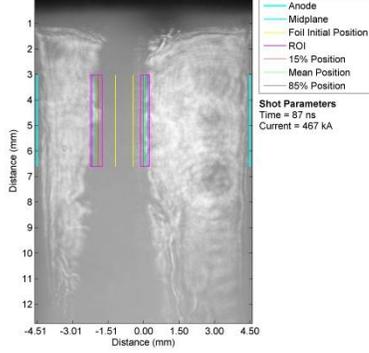
Parallel Profile Lineout | Large Offset | Shot 311 | +/-70 kV



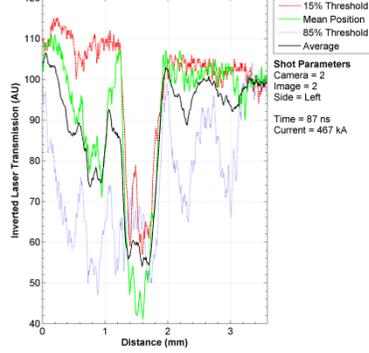
Parallel Profile Lineout | Large Offset | Shot 311 | +/-70 kV



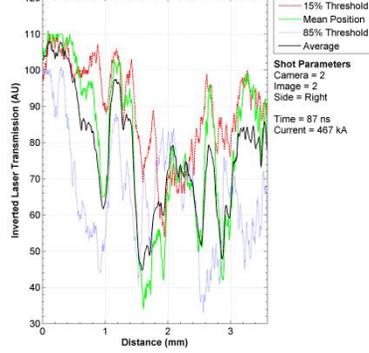
Shot: 314 | Large Offset | +/-70 kV | Camera: 2 | Image: 2



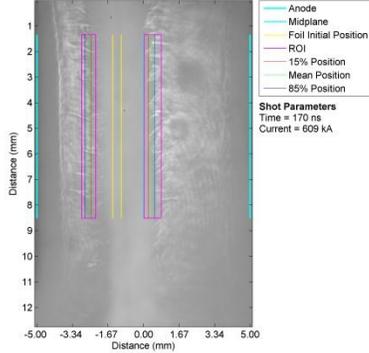
Parallel Profile Lineout | Large Offset | Shot 314 | +/-70 kV



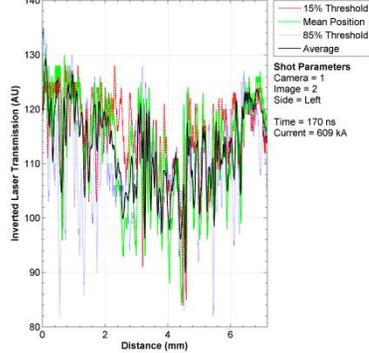
Parallel Profile Lineout | Large Offset | Shot 314 | +/-70 kV



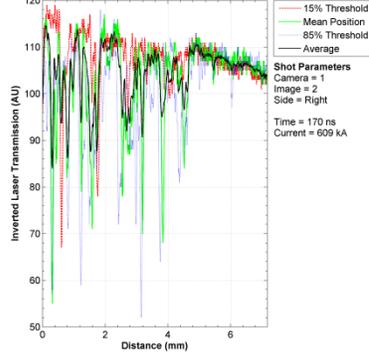
Shot: 337 | Large Offset | +/-70 kV | Camera: 1 | Image: 2



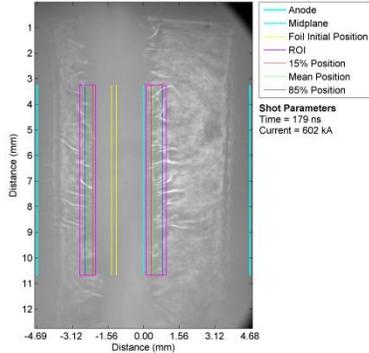
Parallel Profile Lineout | Large Offset | Shot 337 | +/-70 kV



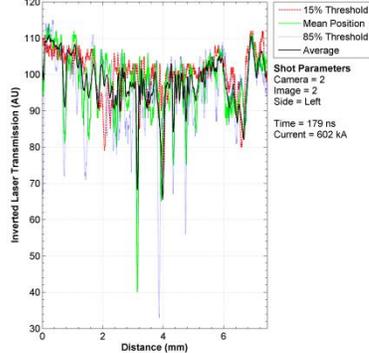
Parallel Profile Lineout | Large Offset | Shot 337 | +/-70 kV



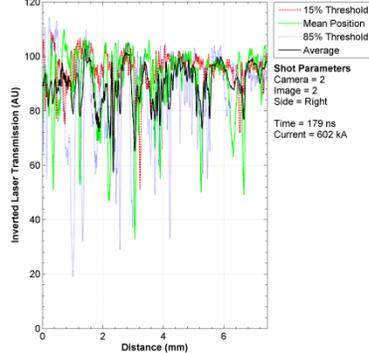
Shot: 337 | Large Offset | +/-70 kV | Camera: 2 | Image: 2



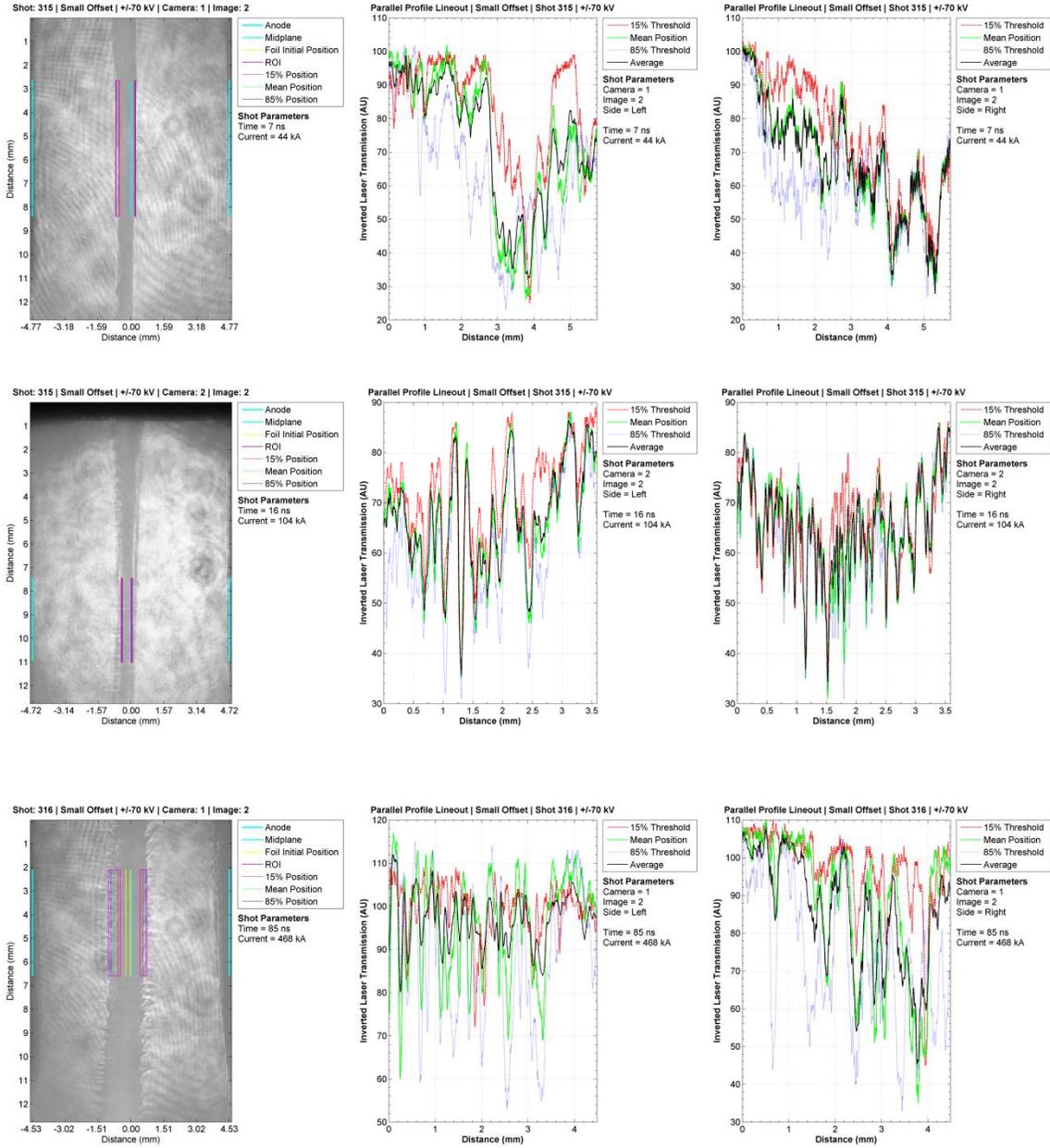
Parallel Profile Lineout | Large Offset | Shot 337 | +/-70 kV



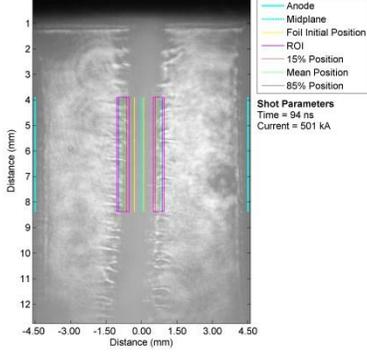
Parallel Profile Lineout | Large Offset | Shot 337 | +/-70 kV



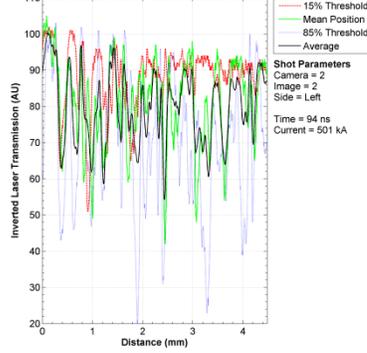
A.4 MRT – Small Foil Offset



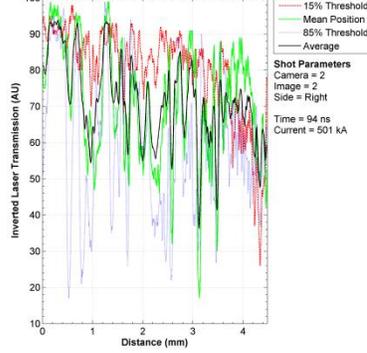
Shot: 316 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



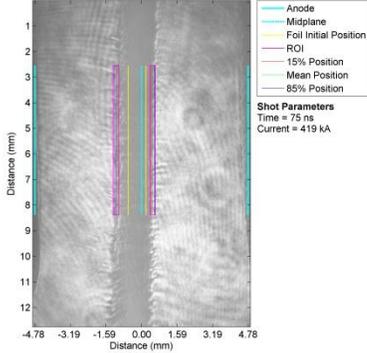
Parallel Profile Lineout | Small Offset | Shot 316 | +/-70 kV



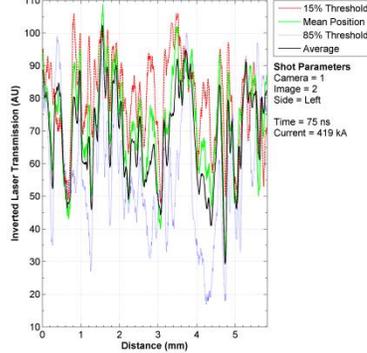
Parallel Profile Lineout | Small Offset | Shot 316 | +/-70 kV



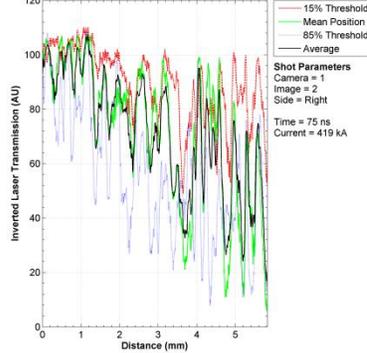
Shot: 319 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



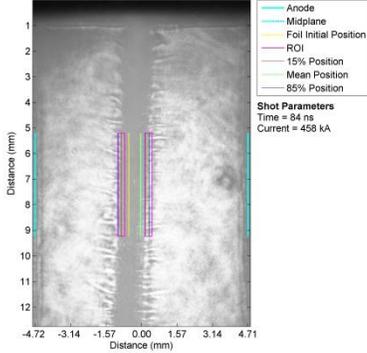
Parallel Profile Lineout | Small Offset | Shot 319 | +/-70 kV



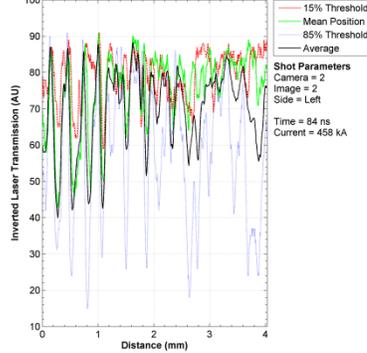
Parallel Profile Lineout | Small Offset | Shot 319 | +/-70 kV



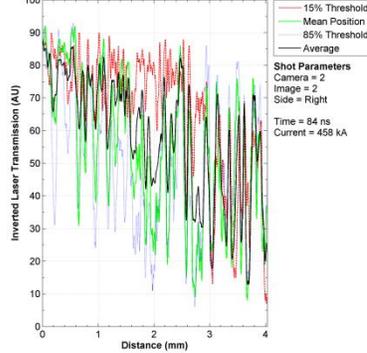
Shot: 319 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



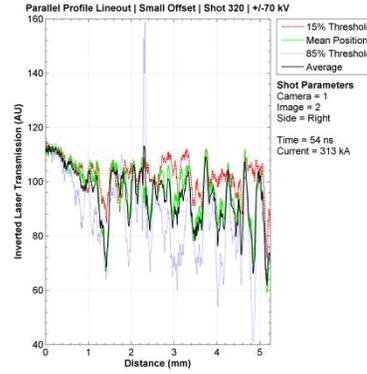
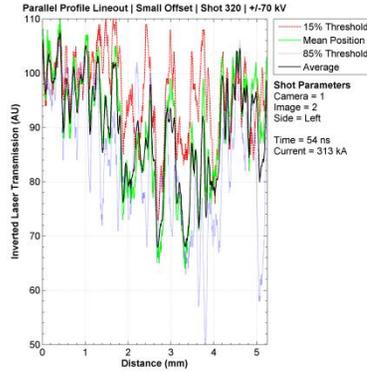
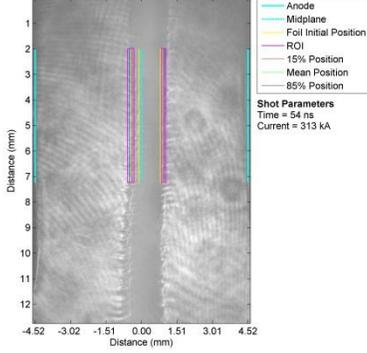
Parallel Profile Lineout | Small Offset | Shot 319 | +/-70 kV



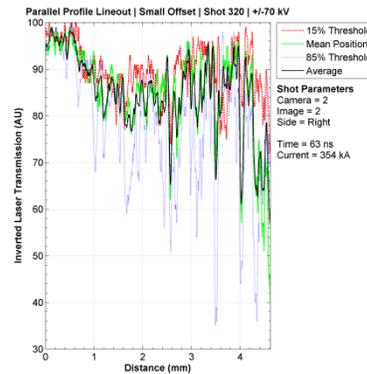
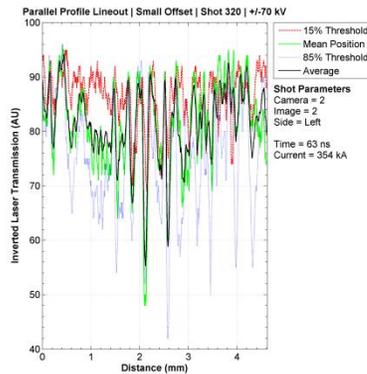
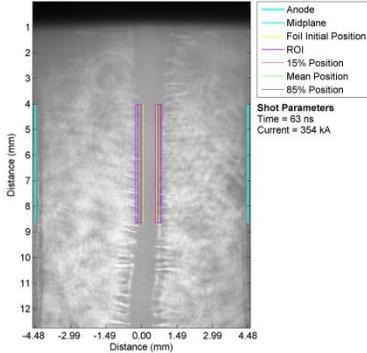
Parallel Profile Lineout | Small Offset | Shot 319 | +/-70 kV



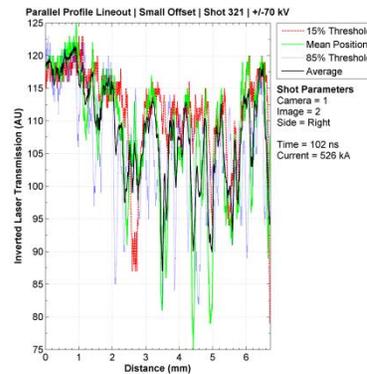
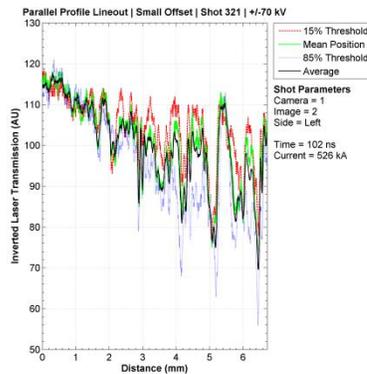
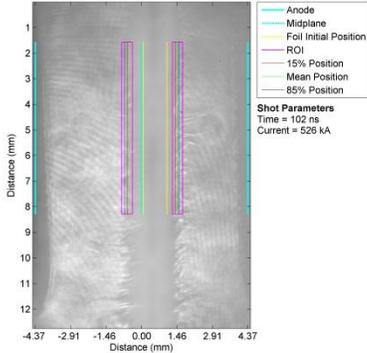
Shot: 320 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



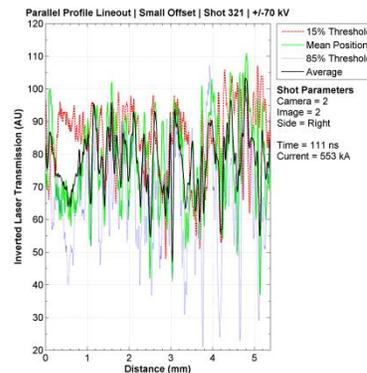
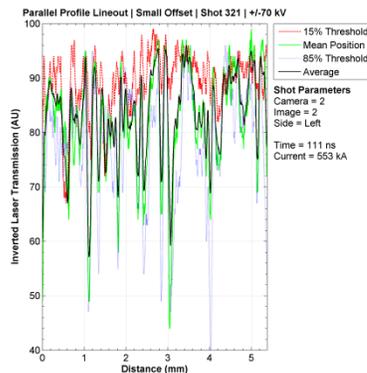
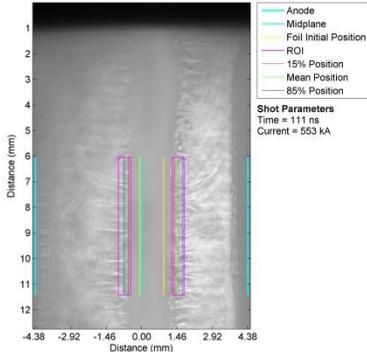
Shot: 320 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



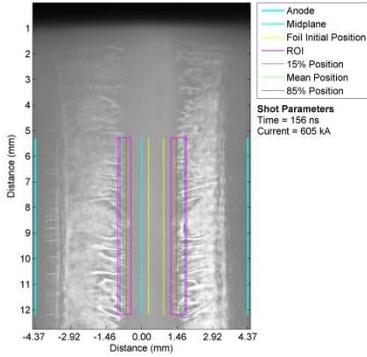
Shot: 321 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



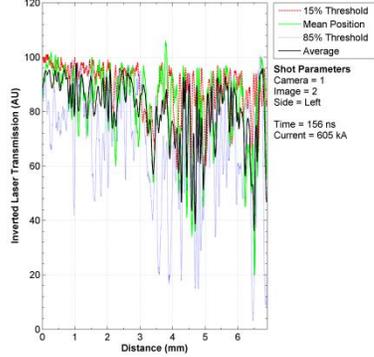
Shot: 321 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



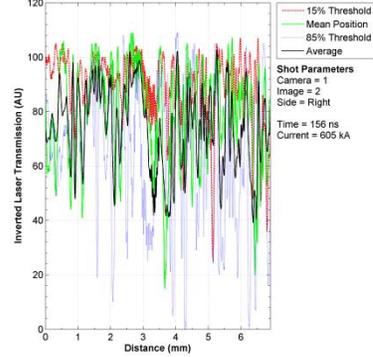
Shot: 322 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



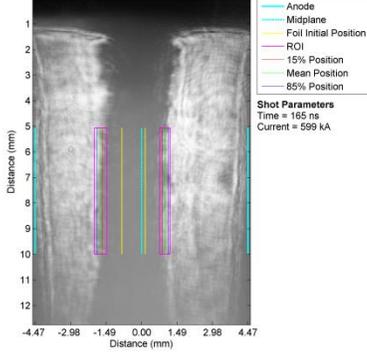
Parallel Profile Lineout | Small Offset | Shot 322 | +/-70 kV



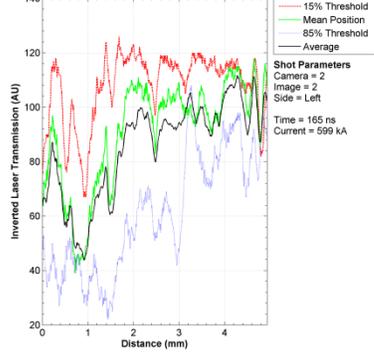
Parallel Profile Lineout | Small Offset | Shot 322 | +/-70 kV



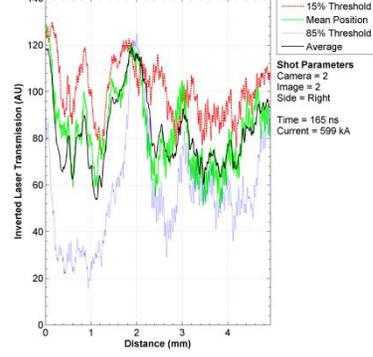
Shot: 322 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



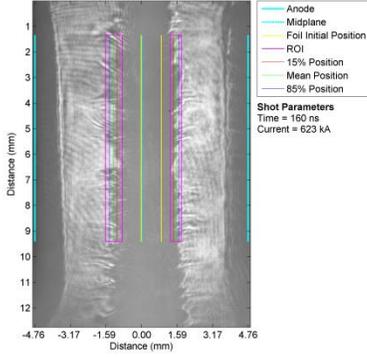
Parallel Profile Lineout | Small Offset | Shot 322 | +/-70 kV



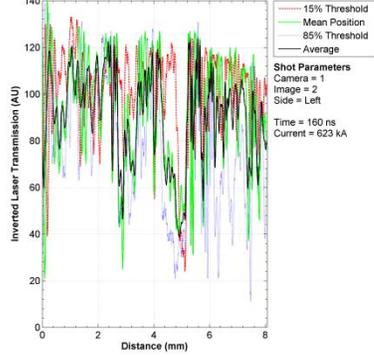
Parallel Profile Lineout | Small Offset | Shot 322 | +/-70 kV



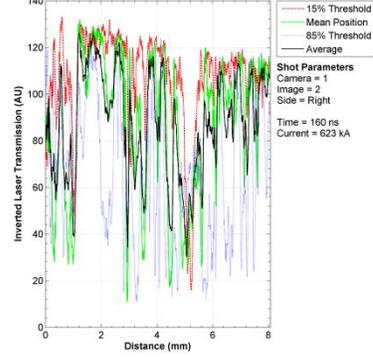
Shot: 323 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



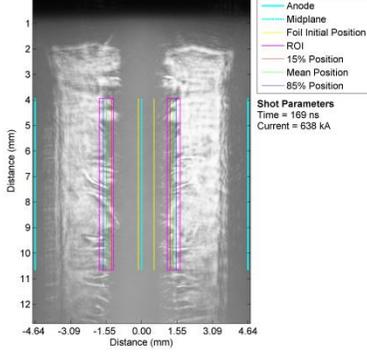
Parallel Profile Lineout | Small Offset | Shot 323 | +/-70 kV



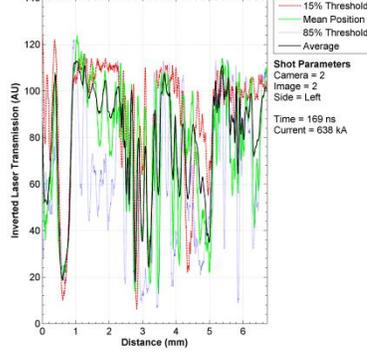
Parallel Profile Lineout | Small Offset | Shot 323 | +/-70 kV



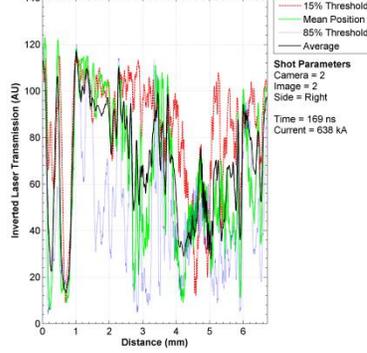
Shot: 323 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



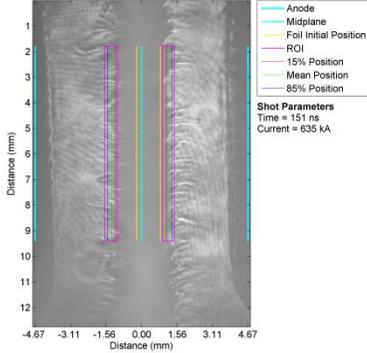
Parallel Profile Lineout | Small Offset | Shot 323 | +/-70 kV



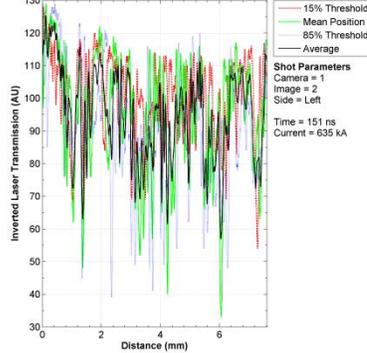
Parallel Profile Lineout | Small Offset | Shot 323 | +/-70 kV



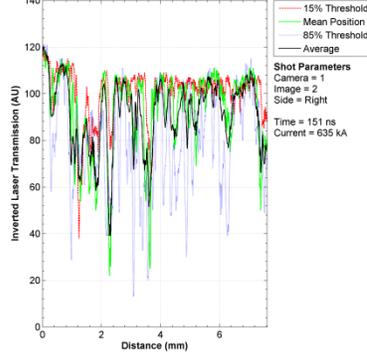
Shot: 324 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



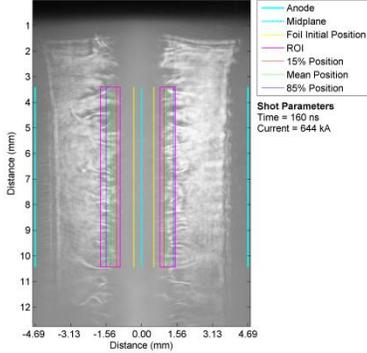
Parallel Profile Lineout | Small Offset | Shot 324 | +/-70 kV



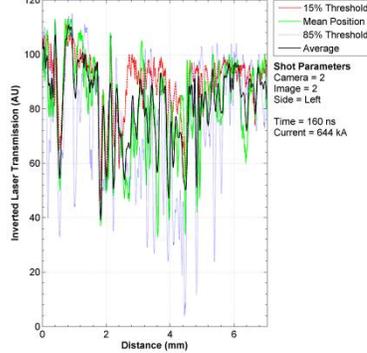
Parallel Profile Lineout | Small Offset | Shot 324 | +/-70 kV



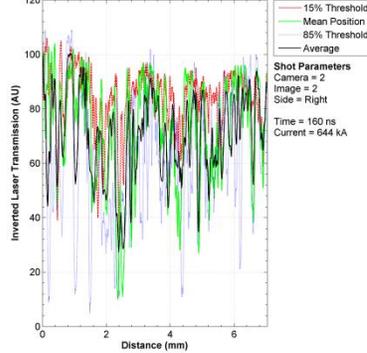
Shot: 324 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



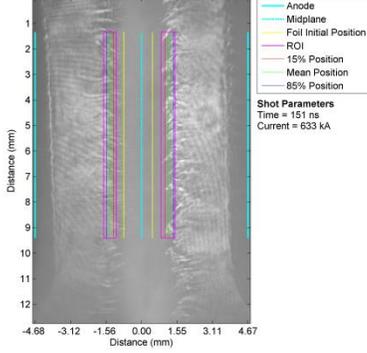
Parallel Profile Lineout | Small Offset | Shot 324 | +/-70 kV



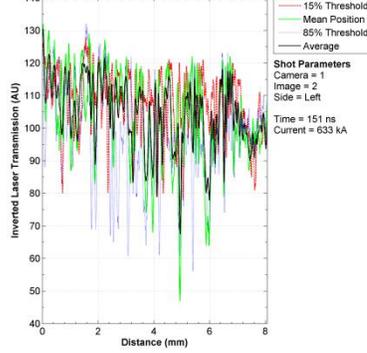
Parallel Profile Lineout | Small Offset | Shot 324 | +/-70 kV



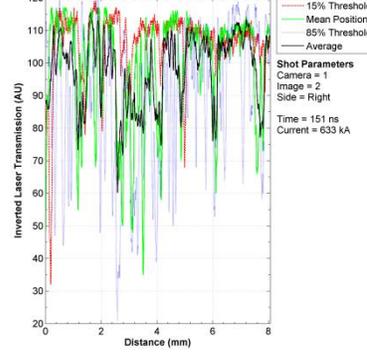
Shot: 325 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



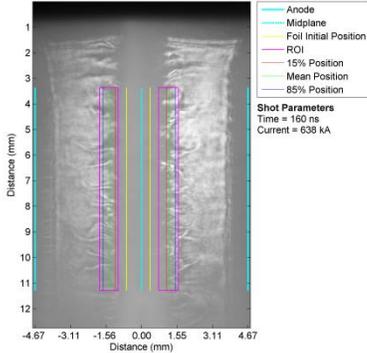
Parallel Profile Lineout | Small Offset | Shot 325 | +/-70 kV



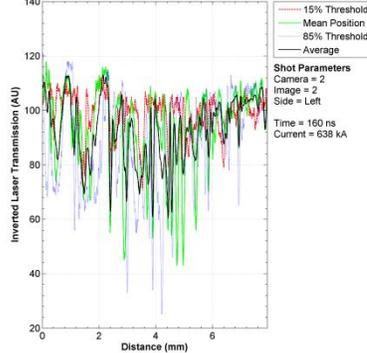
Parallel Profile Lineout | Small Offset | Shot 325 | +/-70 kV



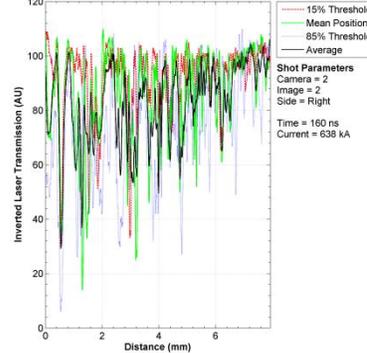
Shot: 325 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



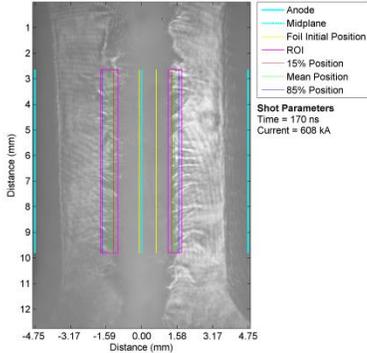
Parallel Profile Lineout | Small Offset | Shot 325 | +/-70 kV



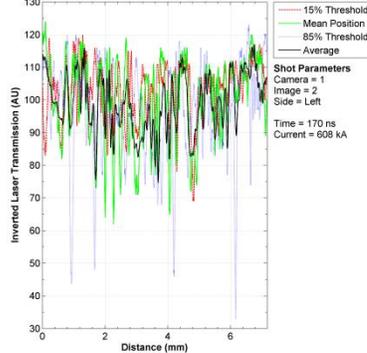
Parallel Profile Lineout | Small Offset | Shot 325 | +/-70 kV



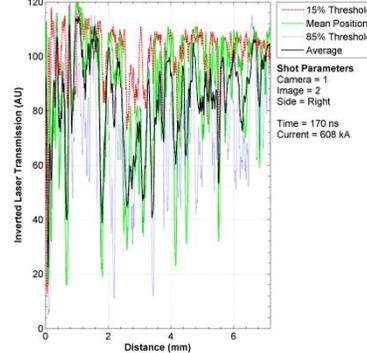
Shot: 326 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



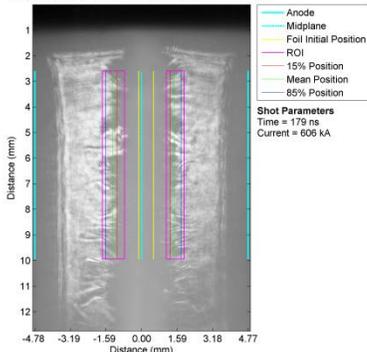
Parallel Profile Lineout | Small Offset | Shot 326 | +/-70 kV



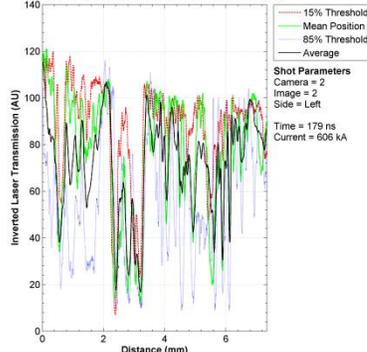
Parallel Profile Lineout | Small Offset | Shot 326 | +/-70 kV



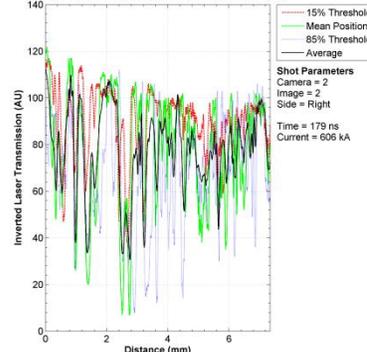
Shot: 326 | Small Offset | +/-70 kV | Camera: 2 | Image: 2

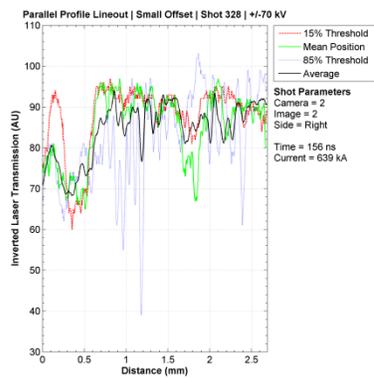
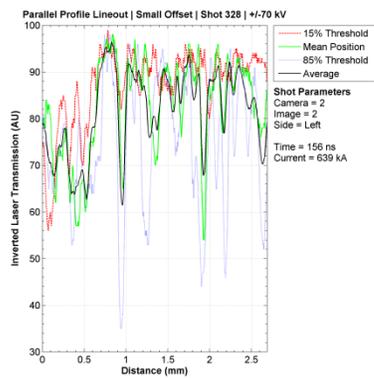
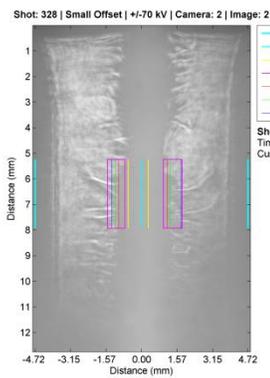
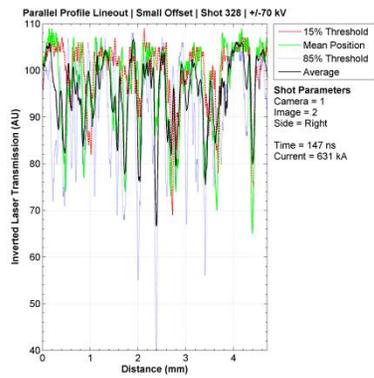
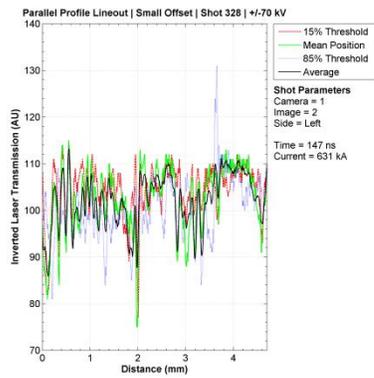
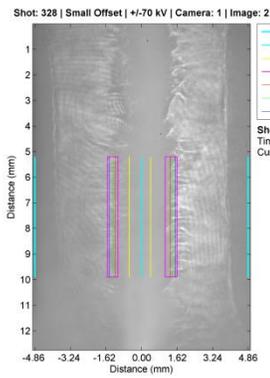


Parallel Profile Lineout | Small Offset | Shot 326 | +/-70 kV

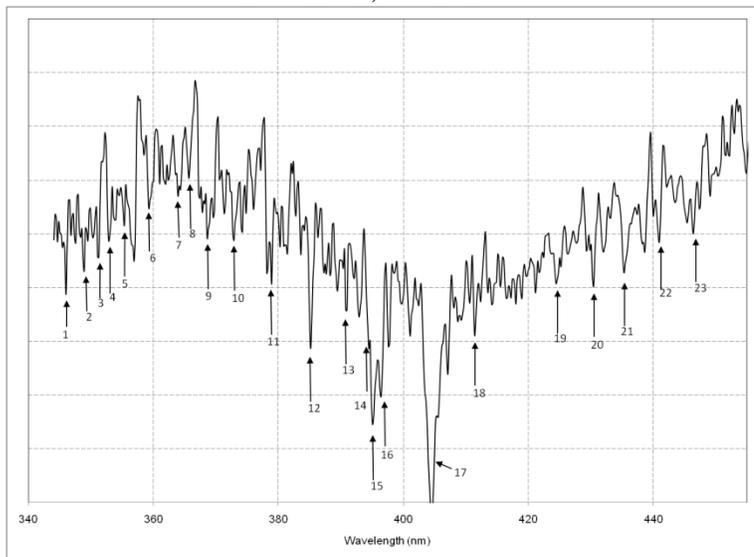


Parallel Profile Lineout | Small Offset | Shot 326 | +/-70 kV



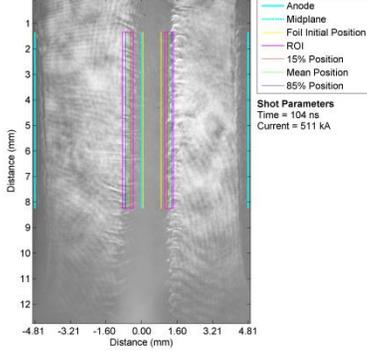


$t = 160 \text{ ns}, I = 640 \text{ kA}$

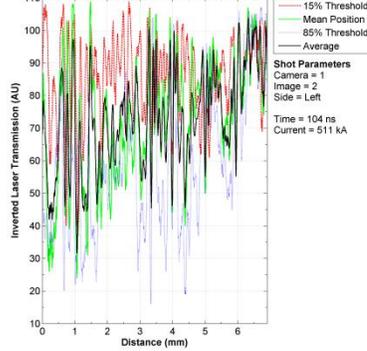


- 1: 346.025 Al II
- 2: 348.945 Al I
- 3: 351.085 Al IV
- 4: 352.837 Al IV
- 5: 355.367 Al II
- 6: 359.258 Al II
- 7: 363.928 Al IV
- 8: 365.679 Al II
- 9: 368.597 Fe I
- 10: 372.876 Al IV
- 11: 378.905 Fe I
- 12: 385.933 Al II
- 13: 390.766 Al II
- 14: 394.460 Al I
- 15: 395.043 Al I
- 16: 396.404 Al I
- 17: 404.373 Fe I
- 18: 411.369 Fe I
- 19: 424.386 Al II
- 20: 430.407 Al II
- 21: 435.263 Al III
- 22: 440.894 Al IV
- 23: 446.885 Al IV

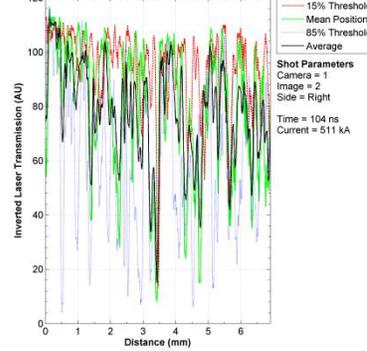
Shot: 331 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



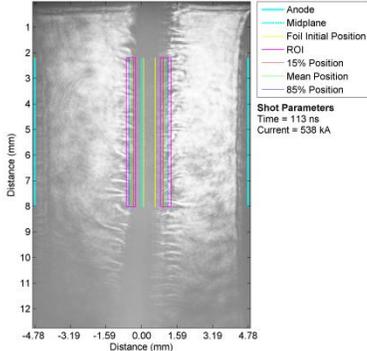
Parallel Profile Lineout | Small Offset | Shot 331 | +/-70 kV



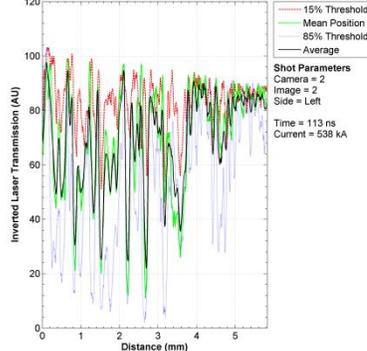
Parallel Profile Lineout | Small Offset | Shot 331 | +/-70 kV



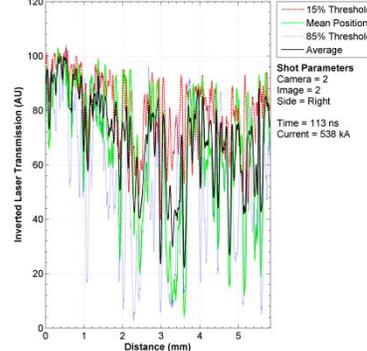
Shot: 331 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



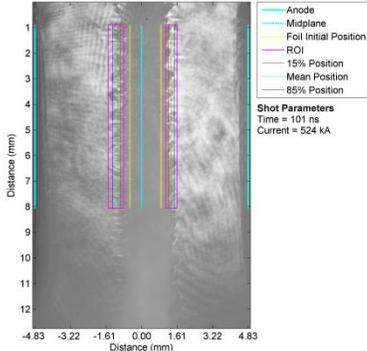
Parallel Profile Lineout | Small Offset | Shot 331 | +/-70 kV



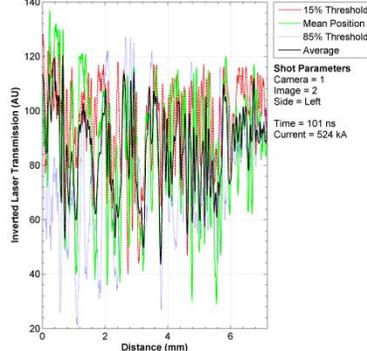
Parallel Profile Lineout | Small Offset | Shot 331 | +/-70 kV



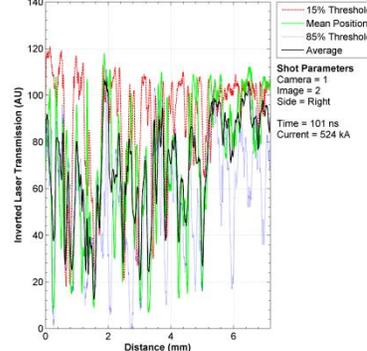
Shot: 332 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



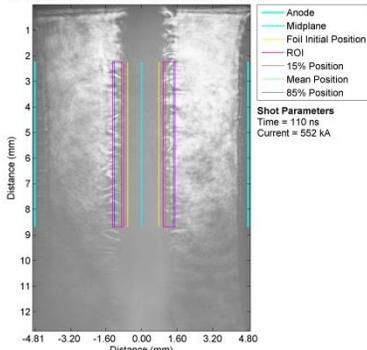
Parallel Profile Lineout | Small Offset | Shot 332 | +/-70 kV



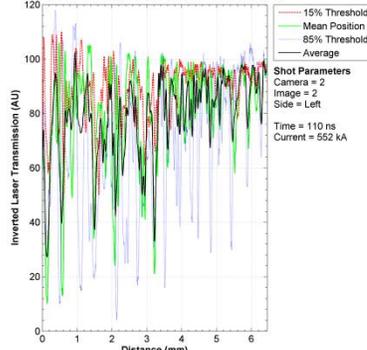
Parallel Profile Lineout | Small Offset | Shot 332 | +/-70 kV



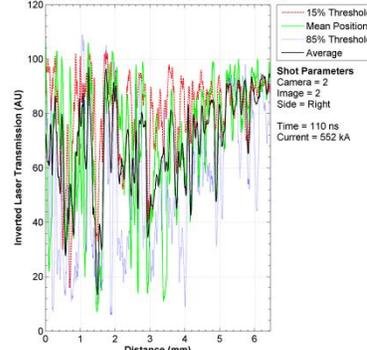
Shot: 332 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



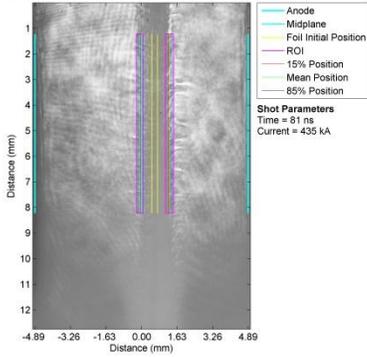
Parallel Profile Lineout | Small Offset | Shot 332 | +/-70 kV



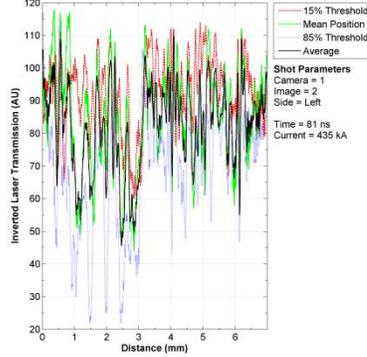
Parallel Profile Lineout | Small Offset | Shot 332 | +/-70 kV



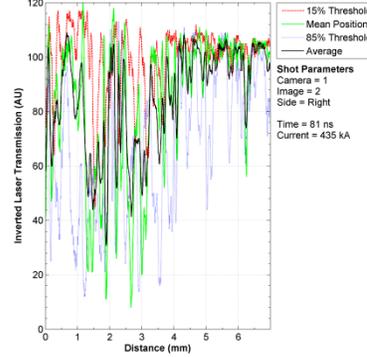
Shot: 333 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



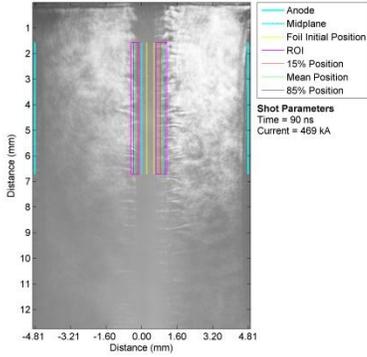
Parallel Profile Lineout | Small Offset | Shot 333 | +/-70 kV



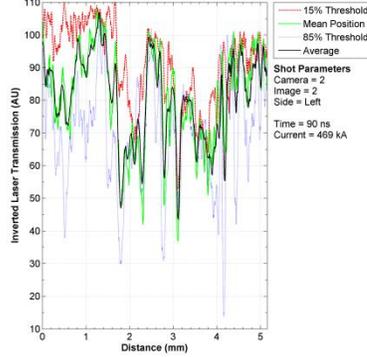
Parallel Profile Lineout | Small Offset | Shot 333 | +/-70 kV



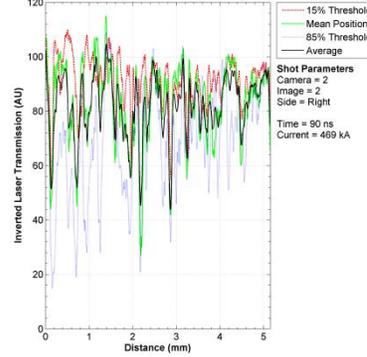
Shot: 333 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



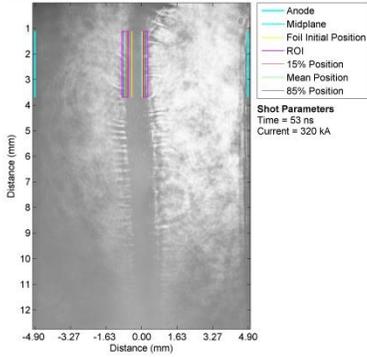
Parallel Profile Lineout | Small Offset | Shot 333 | +/-70 kV



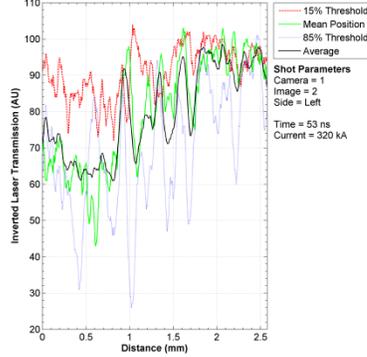
Parallel Profile Lineout | Small Offset | Shot 333 | +/-70 kV



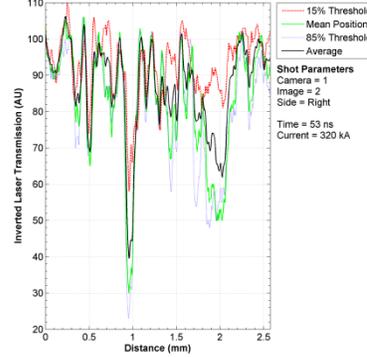
Shot: 334 | Small Offset | +/-70 kV | Camera: 1 | Image: 2



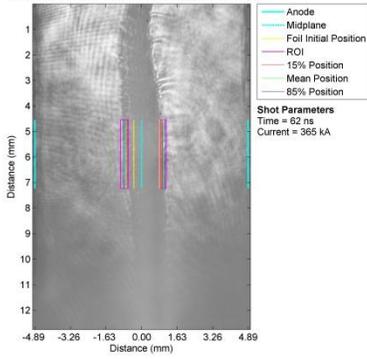
Parallel Profile Lineout | Small Offset | Shot 334 | +/-70 kV



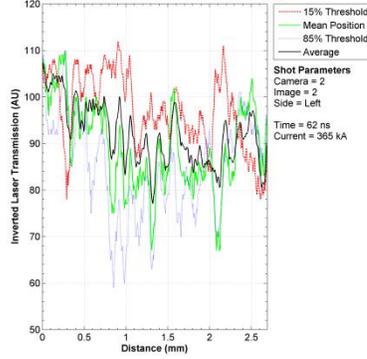
Parallel Profile Lineout | Small Offset | Shot 334 | +/-70 kV



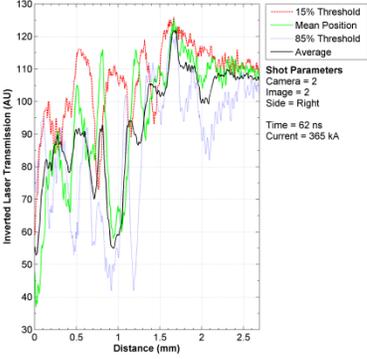
Shot: 334 | Small Offset | +/-70 kV | Camera: 2 | Image: 2



Parallel Profile Lineout | Small Offset | Shot 334 | +/-70 kV



Parallel Profile Lineout | Small Offset | Shot 334 | +/-70 kV



APPENDIX B

MATLAB POSTPROCESSING CODE

B.1 Azimuthally Correlated Ablation

```
% X-Pinch Correlated Ablation Analysis Script
% Jacob Zier
% Date Modified: 03/04/2009
tic
% #####
% User Control Panel

% run options
loadData    = 1;
genLineouts = 1;
analyzeLines = 1;
analyzePairs = 1;
clearOnExit = 0;

% nx1 array of shot numbers to analyze
shots1 = [759;760;761;762;763;765]; % close pairs
shots2 = [785;787;791;794;795;797;798]; % wire diameter sweep
shots3 = [794;795;798;800;802]; % wire spacing sweep

shots = unique([shots1;shots2;shots3;],'rows');
%shots = 762;

% data loading parameters
location = 'C:\data\cobra\shots'; % location of shot number folders

sumloc = 'C:\data\cobra'; % location for summary outputs
sumfile = 'correlation_summary';

pfile = 'xpcorabl'; % parameter file specifier
pdelim = ' '; % delimiter character for traces
imType = 'tif'; % file extension for images

% data filtering parameters
smRange = 5; % lineout smoothing range
smMethod = 'median'; % lineout smoothing method (string)
% 1) 'rloess' : robust 2nd ord. L.Sq.
```

```

% 2) 'median' : median filtering

% pair analysis parameters
pairPer = 2;          % maximum allowed percent difference in magnification

% plot parameters
pltres = 300;
figOutX = 85/12;
figOutY = 85/12;
txtOff = 1/8;

% output options
genAnnImages = 00;
genRawLineout = 00;
  pltRawLineout = 00;
  csvRawLineout = 01;
genPairLineout = 01;
  pltPairLineout = 01;
  csvPairLineout = 00;
csvCorr = 00;

% #####

% ===== start =====
% preliminary calculations

snum = length(shots);
smcut = ceil(smRange/2);

% colors
darkBlue = [0,0,139]/255;
dodgerBlue = [30,144,255]/255;
midnightBlue = [25,25,112]/255;

darkSlateGrey = [47,79,79]/255;
lightSlateGrey = [119,136,153]/255;
grey = [128,128,128]/255;

fireBrick = [178,34,34]/255;
salmon = [250,128,114]/255;
darkRed = [139,0,0]/255;
orangeRed = [255,69,0]/255;

green = [0,128,0]/255;
limeGreen = [50,205,50]/255;
darkGreen = [0,100,0]/255;

% marker order

```

```

markers = {'o','s','d','*','+','x','!','^','v','>','<'};

% ===== end =====

% ===== start =====

% Data Loading
if loadData

display('Loading Data');

% load parameter file
[params,pmap] = load_param(shots,pfile,location);

% load traces
traceID = unique( [vertcat(params(pmap(:,1)).shot),...
                    vertcat(params(pmap(:,1)).scopeID),...
                    vertcat(params(pmap(:,1)).chanID)...
                    ],'rows'...
                );
[traces,tmap] = load_traces(traceID,pdelim,location);
didtID = unique( [vertcat(params(pmap(:,1)).shot),...
                    vertcat(params(pmap(:,1)).scopeID),...
                    vertcat(params(pmap(:,1)).refChanID)...
                    ],'rows'...
                );
[didts,cmap] = load_traces(didtID,pdelim,location);

for i=1:snum
    Inum = length(params(i).shot);
    for j=1:Inum
        pi1 = pmap(i,1);
        logic1 = tmap(:,2)==params(pi1).shot(j);
        logic2 = tmap(:,3)==params(pi1).scopeID(j);
        logic3 = cmap(:,2)==params(pi1).shot(j);
        logic4 = cmap(:,3)==params(pi1).scopeID(j);

        tmap = tmap(logic1 & logic2);
        ltemp = length(tmap);
        for k=1:ltemp
            tk = tmap(k);

            traces(tk).didt = didts(cmap(logic3 & logic4)).signal;
            traces(tk).cTimes = didts(cmap(logic3 & logic4)).times;
            traces(tk).zi = params(pi1).zi(j);

            traces(tk).current = calc_int( ...
                traces(tk).cTimes, ...
                traces(tk).didt, ...

```

```

        traces(tk.zi)';

    end
end

% create image map (converted to load on demand)
imageID = unique([vertcat(params(pmap(:,1)).shot),vertcat(params(pmap(:,1)).imID)],'rows');
imap = [(1:length(imageID(:,1)))',imageID];

% generate wire pairs
imdata(1:length(imap(:,1))) = struct('pairs',[]);
for k=1:length(imap(:,1))
    logic1 = pmap(:,2)==imap(k,2);
    logic2 = params(pmap(logic1,1)).imID==imap(k,3);
    wireInfo = [params(pmap(logic1,1)).wID(logic2),params(pmap(logic1,1)).mag(logic2)];

    imdata(imap(k,1)).pairs = genPairs(wireInfo,pairPer);
end

toc
end

% ===== end =====

% ===== start =====
% Generate Raw Lineouts
if genLineouts

display('Generating Lineouts');

dataID = [vertcat(params(pmap(:,1)).shot),...
          vertcat(params(pmap(:,1)).imID),...
          vertcat(params(pmap(:,1)).wID),...
          vertcat(params(pmap(:,1)).wsID)...
          ];
dmap = [(1:length(dataID(:,1)))',dataID];
for i=1:snum
    pi1 = pmap(i,1);
    lnum = length(params(pi1).shot);
    for j=1:lnum
        logic1 = dmap(:,2)==params(pi1).shot(j);
        logic2 = dmap(:,3)==params(pi1).imID(j);
        logic3 = dmap(:,4)==params(pi1).wID(j);
        logic4 = dmap(:,5)==params(pi1).wsID(j);
        tdmapp = dmap(logic1 & logic2 & logic3 & logic4,1);
        if (params(pi1).wAbl(j)==1)
            data(tdmapp).rwire = NaN;
        end
    end
end

```

```

data(tdmap).rback = NaN;
data(tdmap).wAbl = 1;
else
data(tdmap).wAbl = 0;
logic20 = imap(:,2)==params(pi1).shot(j);
logic21 = imap(:,3)==params(pi1).imID(j);
timap = imap(logic20 & logic21,:);

[image,temp] = load_images(timap(1,2:3),imType.location);

[...
data(tdmap).rwire,...
data(tdmap).rwirestd,...
data(tdmap).rback,...
data(tdmap).rbackstd,...
data(tdmap).xcoords,...
data(tdmap).ycoords...
] = find_wireLineout(...
    image.data,...
    params(pi1).wXL(j),...
    params(pi1).wYL(j),...
    params(pi1).wXR(j),...
    params(pi1).wYR(j),...
    params(pi1).wSch(j),...
    params(pi1).wWidpx(j)...
);

clear image temp;

% lineout smoothing
if strcmpi(smMethod,'median')
    tswire = medfilt1(data(tdmap).rwire,smRange);
    tswirestd = medfilt1(data(tdmap).rwirestd,smRange);
    tsback = medfilt1(data(tdmap).rback,smRange);
    tsbackstd = medfilt1(data(tdmap).rbackstd,smRange);
else
    tswire = smooth(data(tdmap).rwire,smRange,smMethod);
    tswirestd = smooth(data(tdmap).rwirestd,smRange,smMethod);
    tsback = smooth(data(tdmap).rback,smRange,smMethod);
    tsbackstd = smooth(data(tdmap).rbackstd,smRange,smMethod);
end

data(tdmap).swire = tswire(smcut+1:end-smcut);
data(tdmap).swirestd = tswirestd(smcut+1:end-smcut);
data(tdmap).sback = tsback(smcut+1:end-smcut);
data(tdmap).sbackstd = tsbackstd(smcut+1:end-smcut);
data(tdmap).combined = 0;
data(tdmap).mag = params(pi1).mag(j);

```

```

data(tdmap).wXL = params(pi1).wXL(j);
data(tdmap).wXR = params(pi1).wXR(j);
data(tdmap).rdist = [0;cumsum(params(pi1).scale(j).*...
    ones((params(pi1).wXR(j)-params(pi1).wXL(j)),1)/...
    params(pi1).mag(j))];

data(tdmap).len = params(pi1).scale(j)/params(pi1).mag(j)*...
    (params(pi1).wXR(j)-params(pi1).wXL(j));
data(tdmap).smlen = data(tdmap).len - ...
    2*smcut*params(pi1).scale(j)/params(pi1).mag(j);

logic5 = tmap(:,2)==params(pi1).shot(j);
logic6 = tmap(:,3)==params(pi1).scopeID(j);
logic7 = tmap(:,4)==params(pi1).chanID(j);
ttmap = tmap(logic5 & logic6 & logic7);

[ ...
    data(tdmap).time,...
    data(tdmap).sindex...
]= find_peakTime(...
    traces(ttmap).times,...
    traces(ttmap).signal,...
    traces(ttmap).cTimes(traces(ttmap).zi));

ctIndex = find( ...
    traces(ttmap).cTimes(traces(ttmap).zi:end) - ...
    traces(ttmap).cTimes(traces(ttmap).zi) >= ...
    traces(ttmap).times(data(tdmap).sindex) - ...
    traces(ttmap).cTimes(traces(ttmap).zi) ...
    ,1);

data(tdmap).current = ...
    traces(ttmap).current(ctIndex);
data(tdmap).radius = params(pi1).wRadP(j);
data(tdmap).angle = params(pi1).wAngP(j);

end
end

for j=1:lnum
    logic1 = dmap(:,2)==params(pi1).shot(j);
    logic2 = dmap(:,3)==params(pi1).imID(j);
    logic3 = dmap(:,4)==params(pi1).wID(j);
    tdmap = dmap(logic1 & logic2 & logic3,1);
    if (params(pi1).wsID(j)==1 && ...
        params(pi1).wAbl(j)~=1)

        trwire = [];
        trwirestd = [];

```

```

trback = [];
trbackstd = [];
tswire = [];
tswirestd = [];
tsback = [];
tsbackstd = [];
trdist = [];
trlen = [];
trsmilen = [];

for k=1:length(tdmap)
    trwire = [trwire;data(tdmap(k)).rwire];
    trwirestd = [trwirestd;data(tdmap(k)).rwirestd];
    trback = [trback;data(tdmap(k)).rback];
    trbackstd = [trbackstd;data(tdmap(k)).rbackstd];
    tswire = [tswire;data(tdmap(k)).swire];
    tswirestd = [tswirestd;data(tdmap(k)).swirestd];
    tsback = [tsback;data(tdmap(k)).sback];
    tsbackstd = [tsbackstd;data(tdmap(k)).sbackstd];
    trdist = [trdist;data(tdmap(k)).rdist];
    trlen = [trlen;data(tdmap(k)).len];
    trsmilen = [trsmilen;data(tdmap(k)).smilen];
end
logic4 = dmap(:,5)==params(pi1).wsID(j);
tdmap2 = dmap(logic1 & logic2 & logic3 & logic4,1);
tlength = length(dmap(:,1))+1;

data(tlength).rwire = trwire;
data(tlength).rwirestd = trwirestd;
data(tlength).rback = trback;
data(tlength).rbackstd = trbackstd;
data(tlength).swire = tswire;
data(tlength).swirestd = tswirestd;
data(tlength).sback = tsback;
data(tlength).sbackstd = tsbackstd;
data(tlength).xcoords = NaN;
data(tlength).ycoords = NaN;
data(tlength).wXL = NaN;
data(tlength).wXR = NaN;
data(tlength).time = data(tdmap2).time;
% data(tlength).tindex = data(tdmap2).tindex;
data(tlength).sindex = data(tdmap2).sindex;
data(tlength).current = data(tdmap2).current;
data(tlength).wAbl = data(tdmap2).wAbl;
data(tlength).rdist = trdist;

data(tlength).len = sum(trlen);
data(tlength).smilen = sum(trsmilen);

```

```

data(tlength).radius = data(tdmap2).radius;
data(tlength).angle = data(tdmap2).angle;
data(tlength).combined = 1;

dmap = [dmap;tlength,...
        params(pi1).shot(j),...
        params(pi1).imID(j),...
        params(pi1).wID(j),...
        -1];

    end
end
end

toc
end
% ===== end =====

% ===== start =====
% Lineout Analysis
if analyzeLines

display('Analyzing Lineouts');

dnum = length(dmap(:,1));
for i=1:dnum
    if ( data(i).wAbl==0 )

        % #####
        % Lineout analysis operations block

        data(i).df = data(i).swire-data(i).sback;

        swfft = fft(data(i).swire);
        sbfft = fft(data(i).sback);
        dffft = fft(data(i).df);

        [data(i).swfft,data(i).swarg,data(i).swksx] = ...
            fft1 map(data(i).swire,data(i).smlen);
        [data(i).sbfft,data(i).sbarg,data(i).sbksx] = ...
            fft1 map(data(i).sback,data(i).smlen);
        [data(i).dffft,data(i).dfarg,data(i).dfksx] = ...
            fft1 map(data(i).df,data(i).smlen);

        % #####

    end
end
end

```

```

toc
end
% ===== end =====

% ===== start =====
% Pair Analysis
if analyzePairs

display('Analyzing Pairs');

prID = 1;
prmap = zeros(1,5);
dnum = length(dmap(:,1));
for i=1:dnum
    logic1 = dmap(:,2)==dmap(i,2);
    logic2 = dmap(:,3)==dmap(i,3);
    logic3 = dmap(:,5)==dmap(i,5);

    logic20 = imap(:,2)==dmap(i,2);
    logic21 = imap(:,3)==dmap(i,3);
    timap = imap(logic20 & logic21,:);

    logic4 = imdata(timap(1,1)).pairs(:,1)==dmap(i,4);
    timdata = imdata(timap(1,1)).pairs(logic4,:);
    if (sum(logic4)>0)

        for k=1:length(timdata(:,1))

            logic5 = dmap(:,4)==timdata(k,1);
            logic6 = dmap(:,4)==timdata(k,2);
            tmap1 = dmap(logic1 & logic2 & logic3 & logic5);
            tmap2 = dmap(logic1 & logic2 & logic3 & logic6);
            if (~isempty(tmap1) && ...
                ~isempty(tmap2) && ...
                data(tmap1).wAbl==0 && ...
                data(tmap2).wAbl==0 ...
            )

                % #####
                % Pair analysis operations block

                prdata(prID).df = ...
                    (data(tmap1).df-data(tmap2).df);

                % correlation coefficient real space
                [Rsw, Psw, Rlosw, Rhisw] = corrcoef( data(tmap1).swire, data(tmap2).swire);
                [Rsb, Psb, Rlosb, Rhisb] = corrcoef( data(tmap1).sback, data(tmap2).sback);

```

```

[Rdf, Pdf, Rlodf, Rhidf] = corrcoef( data(tdmap1).df, data(tdmap2).df);

prdata(prID).sw_corR = Rsw(1,2);
prdata(prID).sw_corP = Psw(1,2);
prdata(prID).sw_corRlo = Rlosw(1,2);
prdata(prID).sw_corRhi = Rhisw(1,2);

prdata(prID).sb_corR = Rsb(1,2);
prdata(prID).sb_corP = Psb(1,2);
prdata(prID).sb_corRlo = Rlosb(1,2);
prdata(prID).sb_corRhi = Rhisb(1,2);

prdata(prID).df_corR = Rdf(1,2);
prdata(prID).df_corP = Pdf(1,2);
prdata(prID).df_corRlo = Rlodf(1,2);
prdata(prID).df_corRhi = Rhidf(1,2);

% correlation coefficient k-space
[Rswfft, Pswfft, Rloswfft, Rhiswfft] = corrcoef( data(tdmap1).swfft(2:end), data(tdmap2).swfft(2:end));
[Rsbfft, Psbfft, Rlosbfft, Rhisbfft] = corrcoef( data(tdmap1).sbfft(2:end), data(tdmap2).sbfft(2:end));
[Rdffft, Pdffft, Rlodffft, Rhidffft] = corrcoef( data(tdmap1).dffft(2:end), data(tdmap2).dffft(2:end));

prdata(prID).swfft_corR = Rswfft(1,2);
prdata(prID).swfft_corP = Pswfft(1,2);
prdata(prID).swfft_corRlo = Rloswfft(1,2);
prdata(prID).swfft_corRhi = Rhiswfft(1,2);

prdata(prID).sbfft_corR = Rsbfft(1,2);
prdata(prID).sbfft_corP = Psbfft(1,2);
prdata(prID).sbfft_corRlo = Rlosbfft(1,2);
prdata(prID).sbfft_corRhi = Rhisbfft(1,2);

prdata(prID).dffft_corR = Rdffft(1,2);
prdata(prID).dffft_corP = Pdffft(1,2);
prdata(prID).dffft_corRlo = Rlodffft(1,2);
prdata(prID).dffft_corRhi = Rhidffft(1,2);

% correlation coefficient arg-space
[Rswarg, Pswarg, Rloswarg, Rhiswarg] = corrcoef( data(tdmap1).swarg, data(tdmap2).swarg);
[Rsbarg, Psbarg, Rlosbarg, Rhisbarg] = corrcoef( data(tdmap1).sbarg, data(tdmap2).sbarg);
[Rdfarg, Pdfarg, Rlodfarg, Rhidfarg] = corrcoef( data(tdmap1).dfarg, data(tdmap2).dfarg);

prdata(prID).swarg_corR = Rswarg(1,2);
prdata(prID).swarg_corP = Pswarg(1,2);
prdata(prID).swarg_corRlo = Rloswarg(1,2);
prdata(prID).swarg_corRhi = Rhiswarg(1,2);

prdata(prID).sbarg_corR = Rsbarg(1,2);
prdata(prID).sbarg_corP = Psbarg(1,2);

```

```

prdata(prID).sbarg_corRlo = Rlosbarg(1,2);
prdata(prID).sbarg_corRhi = Rhisbarg(1,2);

prdata(prID).dfarg_corR = Rdfarg(1,2);
prdata(prID).dfarg_corP = Pdfarg(1,2);
prdata(prID).dfarg_corRlo = Rlodfarg(1,2);
prdata(prID).dfarg_corRhi = Rhidfarg(1,2);

% #####

tpairIndices = find(logic4);
tpindex=tpairIndices(k);

prmap(prID,1)=prID;
prmap(prID,2)=dmap(i,2);
prmap(prID,3)=dmap(i,3);
prmap(prID,4)=dmap(i,5);
prmap(prID,5)=tpindex;
prID=prID+1;
end
end
end
end

toc
end

% ===== end =====

% ===== start =====
% File Outputs

% annotated images
if (genAnnImages)

display('Generating Annotated Images');

for i=1:snum
pi1 = pmap(i,1);

logic1 = imap(:,2)==pmap(i,2);
timap = imap(logic1,:);
inum = sum(logic1);
for j=1:inum

h=figure('Visible','off');
%h=figure('Visible','on');

```

```

[image,temp] = load_images(timap(j,2:3),imType,location);

imagesc(image.data);
colormap(gray);
hold on;

xmin = length(image.data(1,:));
xmax = 0;

clear image temp;

lnum = length(params(pi1).shot);
for k=1:lnum

    logic2 = dmap(:,2)==timap(j,2);
    logic3 = dmap(:,3)==timap(j,3);
    logic4 = dmap(:,4)==params(pi1).wID(k);
    logic5 = dmap(:,5)==params(pi1).wsID(k);

    tdmapp = dmap(logic2 & logic3 & logic4 & logic5,1);

    if ( data(tdmapp).wAbl~=1 )

        mnum = length(data(tdmapp).ycoords(:,1));
        for m=1:mnum
            plot(data(tdmapp).xcoords,data(tdmapp).ycoords(m,:),...
                'color','g','linewidth',0.1);
        end

        if ( params(pi1).shot(k)==timap(j,2) && ...
            params(pi1).imID(k)==timap(j,3))

            xl=[params(pi1).wXL(k),params(pi1).wXL(k)];
            xt=[params(pi1).wXL(k),params(pi1).wXR(k)];
            xr=[params(pi1).wXR(k),params(pi1).wXR(k)];
            xb=[params(pi1).wXL(k),params(pi1).wXR(k)];
            yl=[params(pi1).wYL(k)+params(pi1).wSch(k),...
                params(pi1).wYL(k)-params(pi1).wSch(k)];
            yt=[params(pi1).wYL(k)+params(pi1).wSch(k),...
                params(pi1).wYR(k)+params(pi1).wSch(k)];
            yr=[params(pi1).wYR(k)+params(pi1).wSch(k),...
                params(pi1).wYR(k)-params(pi1).wSch(k)];
            yb=[params(pi1).wYL(k)-params(pi1).wSch(k),...
                params(pi1).wYR(k)-params(pi1).wSch(k)];

            line(xl,yl,'Color','y','linewidth',0.1);
            line(xt,yt,'Color','y','linewidth',0.1);
            line(xr,yr,'Color','y','linewidth',0.1);
            line(xb,yb,'Color','y','linewidth',0.1);

```

```

line(xl+smcut,yl,'Color','c','linewidth',0.1);
line(xr-smcut,yr,'Color','c','linewidth',0.1);

xmin = min([xmin,min(xl)]);
xmax = max([xmax,max(xr)]);

end
end
end

if xmax==0
close(h);
continue;
end

set(gca, 'FontSize', 16);
xlabel('Pixels');
ylabel('Pixels');
title( [ ...
    '\bfShot: \rm' num2str(timap(j,2)) ...
    ' \bfImage: \rm' num2str(timap(j,3)) ...
    ]);
set(gca,'XMinorTick','on')
set(gca,'YMinorTick','on')

%axis([xmin xmax -inf inf ]);
axis equal;

set(gca, 'Units', 'inches');
set(gca, 'OuterPosition', [0 0 8.5/1.2 8.5/1.2]);
set(gcf, 'PaperPositionMode', 'manual');
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'PaperPosition', [0 0 8.5/1.2 8.5/1.2]);
set(gcf, 'PaperType', 'usletter');
set(gcf, 'ActivePositionProperty', 'outerposition');

filename = [ ...
    'Shot-' num2str(timap(j,2),'%05.0f') ...
    '_Image-' num2str(timap(j,3),'%03.0f') ...
    ];

%print(gcf, '-dttiff', '-r300', filename);
print(gcf, '-dttiff', '-r1000', filename);
%print(gcf, '-dtiffn', filename);

[s,mess,messID] = mkdir( ...
    [location '\ ' num2str(timap(j,2),'%05.0f')...
    'analysis'],'annotated_images');

```

```

if s==1
    movefile( [filename '.tif'], ...
        [location '\' num2str(timap(j,2), '%05.0f') ...
        'analysis\annotated_images']);
else
    display( mess );
    display( num2str(j) );
    break;
end

    hold off;
    close(h);
end
end

toc
end

% raw lineouts
if (genRawLineout)

    dnum = length(dmap(:,1));

    if (pltRawLineout)

        display('Plotting Raw Lineouts');

        for i=1:dnum
            if ( data(i).wAbl==0 && data(i).combined==0 )
                hf1=figure('Visible','off');

                a1p = [0,0,1,1];
                a2p = [0,0,1,1];

                set(gcf,'paperunits','inches');
                set(gcf,'paperposition',[0,0,figOutX,figOutY]);
                set(gcf,'paperpositionmode','auto');
                set(gcf,'PaperSize', [figOutX,figOutY]);
                set(gcf,'units','inches');
                set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
                set(gcf,'units','normalized');

                ha1 = axes(...
                    'Position',a1p,...
                    'Visible','off'...
                );

                ha2 = axes(...
                    'Position',a2p,...

```

```

'activepositionproperty','position',...
'XMinorTick','on',...
'YMinorTick','on',...
'FontSize',11,...
'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Raw Wire','Smooth Wire','Raw Bkgd','Smooth Bkgd'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
dshift = data(i).rdist(smcut+1);
hl1=line( (data(i).rdist(smcut+1:end-smcut)-dshift) *1e3,...
    (data(i).rwire(smcut+1:end-smcut)),'color','k','linewidth',0.5);
    lgdhandles = [lgdhandles,hl1];
hl2=line( (data(i).rdist(smcut+1:end-smcut)-dshift)*1e3,...
    (data(i).swire),'color','limeGreen','linewidth',1.2);
    lgdhandles = [lgdhandles,hl2];
hl5=line( (data(i).rdist(smcut+1:end-smcut)-dshift)*1e3,...
    (data(i).rback(smcut+1:end-smcut)),'color','midnightBlue','linewidth',0.5);
    lgdhandles = [lgdhandles,hl5];
hl6=line( (data(i).rdist(smcut+1:end-smcut)-dshift)*1e3,...
    (data(i).sback),'color','m','linewidth',1.2);
    lgdhandles = [lgdhandles,hl6];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfWire Segment Length (mm)\rm');
ylabel('\bfFilm Response\rm');

title('\bfRaw Wire Lineout\rm');

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontsize',10);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

```

```

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',10);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    ['Shot = ' num2str(dmap(i,2))],...
    ['Image = ' num2str(dmap(i,3))],...
    ['Wire = ' num2str(dmap(i,4))],...
    ['Section = ' num2str(dmap(i,5))],...
    "...
    ['Time = ' num2str(data(i).time*1e9,'%5.1f) ' ns'],...
    ['Current = ' num2str(abs(data(i).current*1e-3),'%6.1f) ' kA]...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...

```

```

    lgdp(3),lgdp(4));
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Shot-' num2str(dmap(i,2),'%05.0f') ...
    '_Image-' num2str(dmap(i,3),'%03.0f') ...
    '_Wire-' num2str(dmap(i,4),'%03.0f') ...
    '_Section-' num2str(dmap(i,5),'%03.0f') ...
    ];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [location '\' num2str(dmap(i,2),'%05.0f')...
    '\analysis\raw_lineouts'],'tif');
if s==1
    movefile( [filename '.tif'], ...
    [location '\' num2str(dmap(i,2),'%05.0f') ...
    '\analysis\raw_lineouts\tif']);
else
    display( mess );
    display( num2str(i) );
    break;
end

close(hf1);
end
end
toc
end

if (csvRawLineout)

display('Writing Raw Lineout CSVs');

for i=1:dnum
    if ( data(i).wAbl==0 && data(i).combined==0 )

        dshift = data(i).rdist(smcut+1);

        headers = ['Distance (m),' ...
            'Raw Wire,' 'Smoothed Wire,' ...
            'Raw Background,' 'Smoothed Background' ...

```

```

];

dataOut = [ ...
    (data(i).rdist(smcut+1:end-smcut)-dshift) ,...
    data(i).rwire(smcut+1:end-smcut)      ,...
    data(i).swire                        ,...
    data(i).rback(smcut+1:end-smcut)     ,...
    data(i).sback                        ...
];

filename = [ ...
    'Shot-' num2str(dmap(i,2),'%05.0f') ...
    '_Image-' num2str(dmap(i,3),'%03.0f') ...
    '_Wire-' num2str(dmap(i,4),'%03.0f') ...
    '_Section-' num2str(dmap(i,5),'%03.0f') ...
    '.csv' ...
];

fid = fopen( filename, 'wt');

if fid==-1
    fclose('all')
    display(['Error writing file ' filename]);
    display( num2str(i) );
    break;
else
    fprintf( fid, '%s\n', headers);
    dcutnum = length(data(i).rdist(smcut+1:end-smcut));
    for k=1:dcutnum;
        fprintf( fid, ...
            '%11.4e,%11.4e,%11.4e,%11.4e,%11.4e\n',...
            dataOut(k,:));
    end
    fclose(fid);

[s,mess,messID] = mkdir( ...
    [location '\' num2str(dmap(i,2),'%05.0f')...
    '\analysis\raw_lineouts'],'csv');
if s
    movefile( filename, ...
        [location '\' num2str(dmap(i,2),'%05.0f') ...
        '\analysis\raw_lineouts\csv']);
else
    display( mess );
    display( num2str(i) );
    break;
end
end
end
end

```

```

        end
    toc
    end
end

% pair lineouts
if (genPairLineout)

    prnum = length(prmap(:,1));

    if (pltPairLineout)

        display('Plotting Pair Lineouts');

        for i=1:prnum

            logic1 = dmap(:,2) == prmap(i,2);
            logic2 = dmap(:,3) == prmap(i,3);
            logic3 = dmap(:,5) == prmap(i,4);

            logic20 = imap(:,2)==prmap(i,2);
            logic21 = imap(:,3)==prmap(i,3);
            timap = imap(logic20 & logic21,:);

            logic4 = dmap(:,4) == imdata(timap(1,1),pairs(prmap(i,5),1));
            logic5 = dmap(:,4) == imdata(timap(1,1),pairs(prmap(i,5),2));
            logic6 = dmap(:,5) ~= -1;

            tdmapp1 = dmap(logic1 & logic2 & logic3 & logic4 & logic6);
            tdmapp2 = dmap(logic1 & logic2 & logic3 & logic5 & logic6);

            if (~isempty(tdmapp1) && ~isempty(tdmapp2))

                hf1=figure('Visible','off');

                a1p = [0,0,1,1];
                a2p = [0,0,1,1];

                set(gcf,'paperunits','inches');
                set(gcf,'paperposition',[0,0,figOutX,figOutY]);
                set(gcf,'paperpositionmode','auto');
                set(gcf,'PaperSize', [figOutX,figOutY]);
                set(gcf,'units','inches');
                set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
                set(gcf,'units','normalized');

                ha1 = axes(...
                    'Position',a1p,...
                    'Visible','off...

```

```

);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',11,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);

dshift1 = data(tdmap1).rdist(smcut+1);
dshift2 = data(tdmap2).rdist(smcut+1);

lgdnames = { ['Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1))], ...
             ['Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2))] ...
             };
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];

hl1=line( (data(tdmap1).rdist(smcut+1:end-smcut)-dshift1)*1e3,...
    (data(tdmap1).df),'color','k','linewidth',0.5);
lgdhandles = [lgdhandles,hl1];
hl4=line( (data(tdmap2).rdist(smcut+1:end-smcut)-dshift2)*1e3,...
    (data(tdmap2).df),'color','r','linewidth',0.5);
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfPair Segment Length (mm)\rm');
ylabel('\bfBackground Corrected Film Response\rm');

title('\bfWire Pair Lineout\rm');

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontsize',10);

```

```

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',10);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    ['Shot = ' num2str(prmap(i,2))],...
    ['Image = ' num2str(prmap(i,3))],...
    ['Pair = ' num2str(prmap(i,5))],...
    ['Section = ' num2str(prmap(i,4))],...
    "...
    ['Time = ' num2str(data(tdmap1).time*1e9,'%5.1f') ' ns'],...
    ['Current = ' num2str(abs(data(tdmap1).current*1e-3),'%6.1f') ' kA']...
    "...
    '\bfCorrelation Coeff.\rm',...
    ['P-Value = ' num2str(prdata(i).df_corP,'%5.3f')],...
    ['R-Value = ' num2str(prdata(i).df_corR,'%5.3f')],...
    ['R-Upper = ' num2str(prdata(i).df_corRhi,'%5.3f')],...
    ['R-Lower = ' num2str(prdata(i).df_corRlo,'%5.3f')],...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...

```

```

        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Shot-' num2str(prmap(i,2),'%05.0f') ...
    '_Image-' num2str(prmap(i,3),'%03.0f') ...
    '_Pair-' num2str(prmap(i,5),'%03.0f') ...
    '_Section-' num2str(prmap(i,4),'%03.0f') ...
    ];

print(gcf, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [location '\' num2str(prmap(i,2),'%05.0f')...
    '\analysis\pair_lineouts'],'tif');
if s==1
    movefile( [filename '.tif'], ...
        [location '\' num2str(prmap(i,2),'%05.0f') ...
        '\analysis\pair_lineouts\tif']);
else
    display( mess );
    display( num2str(i) );
    break;
end

close(hf1);
end
end
toc
end

if (csvPairLineout)

```

```

display('Writing Pair Lineout CSVs');

for i=1:prnum

    logic1 = dmap(:,2) == prmap(i,2);
    logic2 = dmap(:,3) == prmap(i,3);
    logic3 = dmap(:,5) == prmap(i,4);

    logic20 = imap(:,2)==prmap(i,2);
    logic21 = imap(:,3)==prmap(i,3);
    timap = imap(logic20 & logic21,:);

    logic4 = dmap(:,4) == imdata(timap(1,1)).pairs(prmap(i,5),1);
    logic5 = dmap(:,4) == imdata(timap(1,1)).pairs(prmap(i,5),2);
    logic6 = dmap(:,5) ~= -1;

    tdmapp1 = dmap(logic1 & logic2 & logic3 & logic4 & logic6);
    tdmapp2 = dmap(logic1 & logic2 & logic3 & logic5 & logic6);

    if (~isempty(tdmapp1) && ~isempty(tdmapp2))

        dshift1 = data(tdmapp1).rdist(smcut+1);
        dshift2 = data(tdmapp2).rdist(smcut+1);

        headers = [ ...
            'Distance (m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Wr,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Bg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Df,'...
            'Distance (m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Wr,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Bg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Df,'...
            'k (radians/m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Wr Abs,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Bg Abs,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Df Abs,'...
            'k (radians/m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Wr Abs,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Bg Abs,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Df Abs,'...
            'k (radians/m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Wr Arg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Bg Arg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),1)) ' Sm Df Arg,'...
            'k (radians/m),'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Wr Arg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Bg Arg,'...
            'Wire ' num2str(imdata(timap(1,1)).pairs(prmap(i,5),2)) ' Sm Df Arg,'...

```

```

];

dataOut1 = [ ...
    data(tdmap1).rdist(smcut+1:end-smcut)-dshift1    ,...
    data(tdmap1).swire                               ,...
    data(tdmap1).sback                               ,...
    data(tdmap1).df                                  ,...
    data(tdmap2).rdist(smcut+1:end-smcut)-dshift2    ,...
    data(tdmap2).swire                               ,...
    data(tdmap2).sback                               ,...
    data(tdmap2).df                                  ...
];

dataOut2 = [ ...
    data(tdmap1).swksx                               ,...
    data(tdmap1).swfft                               ,...
    data(tdmap1).sbfft                               ,...
    data(tdmap1).dfft                                ,...
    data(tdmap2).swksx                               ,...
    data(tdmap2).swfft                               ,...
    data(tdmap2).sbfft                               ,...
    data(tdmap2).dfft                                ,...
    data(tdmap1).swksx                               ,...
    data(tdmap1).swarg                               ,...
    data(tdmap1).sbarg                               ,...
    data(tdmap1).dfarg                               ,...
    data(tdmap2).swksx                               ,...
    data(tdmap2).swarg                               ,...
    data(tdmap2).sbarg                               ,...
    data(tdmap2).dfarg                               ...
];

filename = [ ...
    'Shot-' num2str(prmap(i,2),'%05.0f') ...
    '_Image-' num2str(prmap(i,3),'%03.0f') ...
    '_Pair-' num2str(prmap(i,5),'%03.0f') ...
    '_Section-' num2str(prmap(i,4),'%03.0f') ...
    '.csv' ...
];

fid = fopen( filename, 'wt');

if fid==-1
    fclose('all')
    display(['Error writing file ' filename]);
    display( num2str(i) );
    break;
else
    fprintf( fid, '%s\n', headers);

```



```

logic1 = dmap(:,2) == prmap(i,2);
logic2 = dmap(:,3) == prmap(i,3);
logic3 = dmap(:,5) == prmap(i,4);

logic20 = imap(:,2)==prmap(i,2);
logic21 = imap(:,3)==prmap(i,3);
timap = imap(logic20 & logic21,:);

logic4 = dmap(:,4) == imdata(timap(1,1)).pairs(prmap(i,5),1);
logic5 = dmap(:,4) == imdata(timap(1,1)).pairs(prmap(i,5),2);

tdmap1 = dmap(logic1 & logic2 & logic3 & logic4);
tdmap2 = dmap(logic1 & logic2 & logic3 & logic5);

tX1 = data(tdmap1).radius*cosd(data(tdmap1).angle);
tX2 = data(tdmap2).radius*cosd(data(tdmap2).angle);
tY1 = data(tdmap1).radius*sind(data(tdmap1).angle);
tY2 = data(tdmap2).radius*sind(data(tdmap2).angle);
tDist = sqrt((tX1-tX2)^2+(tY1-tY2)^2);

dataOut(i,:) = [ ...
    prmap(i,2),...
    prmap(i,3),...
    prmap(i,5),...
    imdata(timap(1,1)).pairs(prmap(i,5),1),...
    imdata(timap(1,1)).pairs(prmap(i,5),2),...
    prmap(i,4),...
    tDist*1e6,...
    data(tdmap1).time*1e9,...
    abs(data(tdmap1).current)*1e-3,...
    prdata(i).sw_corP,...
    prdata(i).sw_corR,...
    prdata(i).sw_corRhi,...
    prdata(i).sw_corRlo, ...
    prdata(i).sb_corP,...
    prdata(i).sb_corR,...
    prdata(i).sb_corRhi,...
    prdata(i).sb_corRlo, ...
    prdata(i).df_corP,...
    prdata(i).df_corR,...
    prdata(i).df_corRhi,...
    prdata(i).df_corRlo, ...
    prdata(i).swfft_corP,...
    prdata(i).swfft_corR,...
    prdata(i).swfft_corRhi,...
    prdata(i).swfft_corRlo, ...
    prdata(i).sbfft_corP,...
    prdata(i).sbfft_corR,...
    prdata(i).sbfft_corRhi,...

```

```

prdata(i).sfft_corRlo, ...
prdata(i).dfft_corP,...
prdata(i).dfft_corR,...
prdata(i).dfft_corRhi,...
prdata(i).dfft_corRlo, ...
prdata(i).swarg_corP,...
prdata(i).swarg_corR,...
prdata(i).swarg_corRhi,...
prdata(i).swarg_corRlo, ...
prdata(i).sbarg_corP,...
prdata(i).sbarg_corR,...
prdata(i).sbarg_corRhi,...
prdata(i).sbarg_corRlo, ...
prdata(i).dfarg_corP,...
prdata(i).dfarg_corR,...
prdata(i).dfarg_corRhi,...
prdata(i).dfarg_corRlo ...
];

end

headers = [ ...
'Shot,' 'Image,' 'Pair,' 'Wire A,' 'Wire B,' ...
'Section,' 'Distance (um),' ... % 'Wire Dia (um),' 'Array Dia (mm),' ...
'Time (ns),' 'Current (kA),' ...
'P-sw,' 'R-sw,' 'Ru-sw,' 'RI-sw,' ...
'P-sb,' 'R-sb,' 'Ru-sb,' 'RI-sb,' ...
'P-df,' 'R-df,' 'Ru-df,' 'RI-df,' ...
'P-swabs,' 'R-swabs,' 'Ru-swabs,' 'RI-swabs,' ...
'P-sbabs,' 'R-sbabs,' 'Ru-sbabs,' 'RI-sbabs,' ...
'P-dfabs,' 'R-dfabs,' 'Ru-dfabs,' 'RI-dfabs,' ...
'P-swarg,' 'R-swarg,' 'Ru-swarg,' 'RI-swarg,' ...
'P-sbarg,' 'R-sbarg,' 'Ru-sbarg,' 'RI-sbarg,' ...
'P-dfarg,' 'R-dfarg,' 'Ru-dfarg,' 'RI-dfarg,' ...
];

filename = [sumfile '.csv'];
fid = fopen( filename, 'wt');

if fid==-1
    fclose('all')
    display(['Error writing file ' filename]);
    display( num2str(i) );
    break;
else
    fprintf( fid, '%s\n', headers);
    fprintf( fid, ...
        ['%11.0f,%11.0f,%11.0f,%11.0f,%11.0f,' ...
        '%11.0f,%11.4e,' ...

```

```

        '%11.4e,%11.4e,' ...
        '%11.4e,%11.4e,%11.4e,%11.4e,' ...
        '%11.4e,%11.4e,%11.4e,%11.4e\n', ...
    dataOut');
fclose(fid);

[s,mess,messID] = mkdir( ...
    [sumloc '\analysis\corr_coeff'],'csv');
if s
    movefile( filename, ...
        [sumloc '\analysis\corr_coeff\csv']);
else
    display( mess );
    display( num2str(i) );
    break;
end
end
toc
end

% ===== end =====
if clearOnExit
    clear;
end
display('*** Analysis Completed ***');
toc

```

B.2 MRT Foils

```

% RT Foils Analysis Script
% Jacob Zier
% Date Modified: 10/29/2010
tic
% #####
% User Control Panel

% run options
loadData    = 01;
exROI      = 01;
clearOnExit = 0;

```

```

% case
seriesData.case = 2;
seriesData.seriesName = {'Large','Small','Excluded','Single'};
% nx1 array of shot numbers to analyze
seriesData.shots = { ...
    [300;301;306;307;308;309;310;311;314;337],...
    [315;316;319;320;321;322;323;324;325;326;328;331;332;333;334],...
    [312;313;317;318;327;330;335;336],...
    327,...
};

% data loading parameters
shotPath = 'C:\data\maize\shots'; % location of shot number folders
paramPath = 'C:\data\maize\parameters'; % location of parameter files
sumPath = 'C:\data\maize\analysis'; % location for summary outputs
distFile = 'Position';
velFile = 'Velocity';
ampFile = 'Amplitude';
thickFile = 'Thickness';
timeFile = 'imageTimes';

pfile = 'foilsOffset'; % parameter file specifier
pdelim = ','; % delimiter character for traces
imType = 'jpg'; % file extension for images

% analysis controls
curTh = 10; % threshold in kA for t=0
streakTh = 0.05; % fraction of max signal for streak timing
specTh = 3e-9; % shift from min signal for spec timing
profileLow = 0.15; % transmission lineout profile thresholds
profileHigh = 0.85;
lineoutAvr = 0; % lineout averaging pixel range
maxLambda = 1.5; % max wavelength for FFT plots (mm)
seriesData.inThBinLow = {0,0,0,0};
seriesData.inThBinHigh = {2,2,2,2};
seriesData.inThBinNum = {15,15,15,15};
seriesData.inPosBinLow = {-2,-1,-2,-2};
seriesData.inPosBinHigh = {0,1,2,2};
seriesData.inPosBinNum = {13,13,13,13};

% nonlinear fit controls
ampUp = 0.8;
tAbl = 38;
fitMethod = 'nonlinearleastsquares';
fitConf = 0.68;
posFitLow = [-5e-2,-5e-4];
posFitHigh = [5e-2,5e-4];
velFitLow = [-1e-2,-1e-4];

```

```

velFitHigh = [1e-2,1e-4];
ampFitLow = [1e-2,-1e-1];
ampFitHigh = [0.5,1e-1];
thCenFitLow = [-1,-1e-2];
thCenFitHigh = [1,1e2];
%posCenFitLow = [-1,-1e-2];
%posCenFitHigh = [1,1e2];
fitIterate = 2000;

% growth rate calculations
rho = 2.7e3;
lfoil = 1e-2;
wvlengthY = 1.0e-3;
wvlengthZ = 1.0e-3;
th = 4e-7;
d0 = 5e-3;
v0 = 0;
mu = 4*pi*1e-7;
seriesData.x0 = {-1e-3,0,-1e-3,-1e-3};

% plot parameters
pltres = 300;
figOutX = 85/12;
figOutY = 85/12;
txtOff = 1/8;
fontSize = 13;
sideText = 0;
plotSpec = 0;

% output options
genPerPro = 0;
genPerProC1L = 01;
genPerProC1R = 01;
genPerProC2L = 01;
genPerProC2R = 01;

genParPro = 0;
genParProC1L = 01;
genParProC1R = 01;
genParProC2L = 01;
genParProC2R = 01;
genParKProC1L = 01;
genParKProC1R = 01;
genParKProC2L = 01;
genParKProC2R = 01;

genBulkKinematics = 01;
genPosLeft = 0;
genPosRight = 0;

```

```

genVelLeft    = 0;
genVelRight   = 0;
genAmpLeft    = 0;
genAmpRight   = 0;
genInitialTh  = 01;
genInitialPos = 01;
genInstantTh  = 01;
genInstantPos = 01;

genAnnImages  = 0;

genTimePlot   = 0;
genSingleTime = 01;
genSumTime    = 01;

genDynamicPlots = 0;
genKinematics  = 0;
genMagFields   = 01;
genGrowthRates = 0;

% #####

% ===== start =====
% preliminary calculations

series      = seriesData.seriesName{seriesData.case};
shots      = seriesData.shots{seriesData.case};
inThBinLow = seriesData.inThBinLow{seriesData.case};
inThBinHigh = seriesData.inThBinHigh{seriesData.case};
inThBinNum = seriesData.inThBinNum{seriesData.case};
inPosBinLow = seriesData.inPosBinLow{seriesData.case};
inPosBinHigh = seriesData.inPosBinHigh{seriesData.case};
inPosBinNum = seriesData.inPosBinNum{seriesData.case};
x0         = seriesData.x0{seriesData.case};

snum = length(shots);

% colors
darkBlue    = [0,0,139]/255;
dodgerBlue  = [30,144,255]/255;
midnightBlue = [25,25,112]/255;
lightCyan   = [200,255,255]/255;

darkSlateGrey = [47,79,79]/255;
lightSlateGrey = [119,136,153]/255;
grey         = [128,128,128]/255;

fireBrick   = [178,34,34]/255;

```

```

salmon      = [250,128,114]/255;
darkRed     = [139,0,0]/255;
orangeRed   = [255,69,0]/255;

green       = [0,128,0]/255;
limeGreen   = [50,205,50]/255;
darkGreen   = [0,100,0]/255;

% marker order
markers = {'o','s','d','*','+','x','!','^','v','>','<'};

% ===== end =====

% ===== start =====

% Data Loading
if loadData

display('Loading Data');

% load parameter file
[params,pmap] = load_paramSingle(shots,pfile,paramPath);

% load traces
traceID1 = unique( [vertcat(params.shot),...
                    vertcat(params.bdot2ScopeID),...
                    vertcat(params.bdot2Chan1ID)...
                    ],'rows'...
                  );
traceID2 = unique( [vertcat(params.shot),...
                    vertcat(params.bdot2ScopeID),...
                    vertcat(params.bdot2Chan2ID)...
                    ],'rows'...
                  );
traceID3 = unique( [vertcat(params.shot),...
                    vertcat(params.laserScopeID),...
                    vertcat(params.laserChanID)...
                    ],'rows'...
                  );
traceID4 = unique( [vertcat(params.shot),...
                    vertcat(params.specScopeID),...
                    vertcat(params.specChanID)...
                    ],'rows'...
                  );
traceID5 = unique( [vertcat(params.shot),...
                    vertcat(params.streakScopeID),...
                    vertcat(params.streakChanID)...
                    ],'rows'...
                  );

```

```

traceID = unique([traceID1;traceID2;traceID3;traceID4;traceID5;],'rows');

[traces,tmap] = load_traces(traceID,pdelim,shotPath);

for k=1:snum
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = tmap(:,2)==params.shot(logicP);
    logic2 = tmap(:,3)==params.bdot2ScopeID(logicP);
    logic3 = tmap(:,4)==params.bdot2Chan1ID(logicP);
    logic4 = tmap(:,4)==params.bdot2Chan2ID(logicP);

    tmap1 = tmap(logic1 & logic2 & logic3);
    tmap2 = tmap(logic1 & logic2 & logic4);
    ltemp1 = length(tmap1);
    ltemp2 = length(tmap2);

    sdata(pm1).current = calc_intz( ...
        traces(tmap1).times, ...
        (traces(tmap1).signal-traces(tmap2).signal)*params.bdot2Cal(logicP, ...
        params.bdot2ZI(logicP));
    sdata(pm1).curTh = find(sdata(pm1).current>=curTh,1,'first');
    sdata(pm1).cShift = traces(tmap1).times(sdata(pm1).curTh);
    sdata(pm1).cTimes = traces(tmap1).times-sdata(pm1).cShift;
    [sdata(pm1).cMax,sdata(pm1).cMaxI] = max(sdata(pm1).current);
    sdata(pm1).cMaxT = traces(tmap1).times(sdata(pm1).cMaxI) - ...
        sdata(pm1).cShift;

    logic5 = tmap(:,3)==params.laserScopeID(logicP);
    logic6 = tmap(:,4)==params.laserChanID(logicP);
    [~,sdata(pm1).laserI] = max(traces(logic1 & logic5 & logic6).signal);
    sdata(pm1).cam1Time = traces(logic1 & logic5 & logic6).times(sdata(pm1).laserI) - ...
        params.laser1Delay(logicP) + params.bdot2Delay(logicP) - ...
        sdata(pm1).cShift;
    sdata(pm1).cam1CurInd = find(sdata(pm1).cTimes>=sdata(pm1).cam1Time,1,'first');
    sdata(pm1).cam1Cur = sdata(pm1).current(sdata(pm1).cam1CurInd);

    sdata(pm1).cam2Time = traces(logic1 & logic5 & logic6).times(sdata(pm1).laserI) - ...
        params.laser2Delay(logicP) + params.bdot2Delay(logicP) - ...
        sdata(pm1).cShift;
    sdata(pm1).cam2CurInd = find(sdata(pm1).cTimes>=sdata(pm1).cam2Time,1,'first');
    sdata(pm1).cam2Cur = sdata(pm1).current(sdata(pm1).cam2CurInd);

    logic7 = tmap(:,3)==params.streakScopeID(logicP);
    logic8 = tmap(:,4)==params.streakChanID(logicP);
    sdata(pm1).streakI = find(traces(logic1 & logic7 & logic8).signal>= ...
        streakTh*max(traces(logic1 & logic7 & logic8).signal),1,'first');
    sdata(pm1).streakTime = traces(logic1 & logic7 & logic8).times(sdata(pm1).streakI) - ...
        params.streakDelay(logicP) + params.bdot2Delay(logicP) - ...

```

```

        sdata(pm1).cShift;

    logic9 = tmap(:,3)==params.specScopeID(logicP);
    logic10 = tmap(:,4)==params.specChanID(logicP);
    [~,temp] = min(traces(logic1 & logic9 & logic10).signal);
    sdata(pm1).specI = find(traces(logic1 & logic9 & logic10).times >= ...
        traces(logic1 & logic9 & logic10).times(temp)-specTh,1,'first');
    sdata(pm1).specTime = traces(logic1 & logic9 & logic10).times(sdata(pm1).specI)- ...
        params.specDelay(logicP) + params.bdot2Delay(logicP) - ...
        sdata(pm1).cShift;

end

% create image map (converted to load on demand)
imageID1 = unique( [vertcat(params.shot),...
    vertcat(params.cam1ID),...
    vertcat(params.cam1PreID)...
    ],'rows'...
);
imageID2 = unique( [vertcat(params.shot),...
    vertcat(params.cam1ID),...
    vertcat(params.cam1ImID)...
    ],'rows'...
);
imageID3 = unique( [vertcat(params.shot),...
    vertcat(params.cam2ID),...
    vertcat(params.cam2PreID)...
    ],'rows'...
);
imageID4 = unique( [vertcat(params.shot),...
    vertcat(params.cam2ID),...
    vertcat(params.cam2ImID)...
    ],'rows'...
);
imageID5 = unique( [vertcat(params.shot),...
    vertcat(params.cam3ID),...
    vertcat(params.cam3ImID)...
    ],'rows'...
);
imageID = unique([imageID1;imageID2;imageID3;imageID4;imageID5;],'rows');
imap = [(1:length(imageID(:,1)))',imageID];

toc
end
% ===== end =====

% ===== start =====
% Extract Regions of Interest

```

```

if exROI
display('Extracting Regions of Interest');

for k=1:snum

    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    % Camera 1
    logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
    if sum(logicP1) > 0
        logic2 = imap(:,3)==params.cam1ID(logicP1);
        logic3 = imap(:,4)==params.cam1PreID(logicP1);
        logic4 = imap(:,4)==params.cam1ImID(logicP1);
        timap = imap(logic1 & logic2 & logic4,:);

        [image,temp] = load_images(timap(1,2:4),imType,shotPath);

        sdata(pm1).c1pxl = params.cam1pxl(logicP1);
        sdata(pm1).c1pxr = sdata(pm1).c1pxl + params.cam1pxld(logicP1);
        sdata(pm1).c1pfxl = params.cam1pfl(logicP1);
        sdata(pm1).c1pfxr = sdata(pm1).c1pfxl + params.cam1pfld(logicP1);
        sdata(pm1).c1yt = params.cam1yt(logicP1);
        sdata(pm1).c1yb = params.cam1yt(logicP1)+params.cam1yd(logicP1);
        sdata(pm1).c1lxl = params.cam1xl(logicP1);
        sdata(pm1).c1lxr = params.cam1x2(logicP1);
        sdata(pm1).c1rxl = params.cam1x3(logicP1);
        sdata(pm1).c1rxr = params.cam1x4(logicP1);

        roiL = image.data(sdata(pm1).c1yt:sdata(pm1).c1yb,sdata(pm1).c1lxl:sdata(pm1).c1lxr,2);
        roiR = image.data(sdata(pm1).c1yt:sdata(pm1).c1yb,sdata(pm1).c1rxl:sdata(pm1).c1rxr,2);

        sdata(pm1).c1profileIntL = mean(roiL,1);
        sdata(pm1).c1profileIntR = mean(roiR,1);

        sdata(pm1).c1MidDistInd = round((sdata(pm1).c1pxr + sdata(pm1).c1pxl)/2);
        sdata(pm1).c1profileDistL = ((sdata(pm1).c1lxl:sdata(pm1).c1lxr)-sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
        sdata(pm1).c1profileDistR = ((sdata(pm1).c1rxl:sdata(pm1).c1rxr)-sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
        sdata(pm1).c1lineoutDist = ((sdata(pm1).c1yt:sdata(pm1).c1yb) -sdata(pm1).c1yt)*params.cam1Scale(logicP1);

        sdata(pm1).c1profileLowIntL = (max(sdata(pm1).c1profileIntL)-min(sdata(pm1).c1profileIntL))*profileLow
+min(sdata(pm1).c1profileIntL);
        sdata(pm1).c1profileMidIntL = mean(sdata(pm1).c1profileIntL);
        sdata(pm1).c1profileHighIntL = (max(sdata(pm1).c1profileIntL)-
min(sdata(pm1).c1profileIntL))*profileHigh+min(sdata(pm1).c1profileIntL);
        sdata(pm1).c1profileLowIndL = find(sdata(pm1).c1profileIntL >= sdata(pm1).c1profileLowIntL ,1,'last');
        sdata(pm1).c1profileMidIndL1 = find(sdata(pm1).c1profileIntL >= sdata(pm1).c1profileMidIntL ,1,'last');
        sdata(pm1).c1profileMidIndL2 = find(sdata(pm1).c1profileIntL <= sdata(pm1).c1profileMidIntL ,1,'first');
        sdata(pm1).c1profileMidIndL = round(mean([sdata(pm1).c1profileMidIndL1,sdata(pm1).c1profileMidIndL2]));

```

```

sdata(pm1).c1profileHighIndL = find(sdata(pm1).c1profileIntL <= sdata(pm1).c1profileHighIntL,1,'first');

% sdata(pm1).c1profileMidIndL = round((sdata(pm1).c1profileHighIndL-
sdata(pm1).c1profileLowIndL)/2+sdata(pm1).c1profileLowIndL);
sdata(pm1).c1profileLowIntL = sdata(pm1).c1profileIntL(sdata(pm1).c1profileLowIndL);
sdata(pm1).c1profileMidIntL = sdata(pm1).c1profileIntL(sdata(pm1).c1profileMidIndL);
sdata(pm1).c1profileHighIntL = sdata(pm1).c1profileIntL(sdata(pm1).c1profileHighIndL);

sdata(pm1).c1lineoutLowL = 255-mean(roiL(:,sdata(pm1).c1profileLowIndL -lineoutAvr:sdata(pm1).c1profileLowIndL
+lineoutAvr),2);
sdata(pm1).c1lineoutMidL = 255-mean(roiL(:,sdata(pm1).c1profileMidIndL -lineoutAvr:sdata(pm1).c1profileMidIndL
+lineoutAvr),2);
sdata(pm1).c1lineoutHighL = 255-mean(roiL(:,sdata(pm1).c1profileHighIndL-
lineoutAvr:sdata(pm1).c1profileHighIndL+lineoutAvr),2);
sdata(pm1).c1lineoutTotL = 255-mean(roiL(:,sdata(pm1).c1profileHighIndL:sdata(pm1).c1profileLowIndL),2);

sdata(pm1).c1profileLowIntR = (max(sdata(pm1).c1profileIntR)-min(sdata(pm1).c1profileIntR))*profileLow
+min(sdata(pm1).c1profileIntR);
sdata(pm1).c1profileMidIntR = mean(sdata(pm1).c1profileIntR);
sdata(pm1).c1profileHighIntR = (max(sdata(pm1).c1profileIntR)-
min(sdata(pm1).c1profileIntR))*profileHigh+min(sdata(pm1).c1profileIntR);
sdata(pm1).c1profileLowIndR = find(sdata(pm1).c1profileIntR >= sdata(pm1).c1profileLowIntR ,1,'first');
sdata(pm1).c1profileMidIndR1 = find(sdata(pm1).c1profileIntR >= sdata(pm1).c1profileMidIntR ,1,'first');
sdata(pm1).c1profileMidIndR2 = find(sdata(pm1).c1profileIntR <= sdata(pm1).c1profileMidIntR ,1,'last');
sdata(pm1).c1profileMidIndR = round(mean([sdata(pm1).c1profileMidIndR1,sdata(pm1).c1profileMidIndR2]));
sdata(pm1).c1profileHighIndR = find(sdata(pm1).c1profileIntR <= sdata(pm1).c1profileHighIntR,1,'last');

% sdata(pm1).c1profileMidIndR = round((sdata(pm1).c1profileHighIndR-
sdata(pm1).c1profileLowIndR)/2+sdata(pm1).c1profileLowIndR);
sdata(pm1).c1profileLowIntR = sdata(pm1).c1profileIntR(sdata(pm1).c1profileLowIndR);
sdata(pm1).c1profileMidIntR = sdata(pm1).c1profileIntR(sdata(pm1).c1profileMidIndR);
sdata(pm1).c1profileHighIntR = sdata(pm1).c1profileIntR(sdata(pm1).c1profileHighIndR);

sdata(pm1).c1lineoutLowR = 255-mean(roiR(:,sdata(pm1).c1profileLowIndR -lineoutAvr:sdata(pm1).c1profileLowIndR
+lineoutAvr),2);
sdata(pm1).c1lineoutMidR = 255-mean(roiR(:,sdata(pm1).c1profileMidIndR -lineoutAvr:sdata(pm1).c1profileMidIndR
+lineoutAvr),2);
sdata(pm1).c1lineoutHighR = 255-mean(roiR(:,sdata(pm1).c1profileHighIndR-
lineoutAvr:sdata(pm1).c1profileHighIndR+lineoutAvr),2);
sdata(pm1).c1lineoutTotR = 255-mean(roiR(:,sdata(pm1).c1profileLowIndR:sdata(pm1).c1profileHighIndR),2);

% perpendicular calculaions
sdata(pm1).c1LowL = (sdata(pm1).c1profileLowIndL + sdata(pm1).c1lxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
sdata(pm1).c1MidL = (sdata(pm1).c1profileMidIndL + sdata(pm1).c1lxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
sdata(pm1).c1HighL = (sdata(pm1).c1profileHighIndL + sdata(pm1).c1lxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);

sdata(pm1).c1LowR = (sdata(pm1).c1profileLowIndR + sdata(pm1).c1lxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
sdata(pm1).c1MidR = (sdata(pm1).c1profileMidIndR + sdata(pm1).c1lxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);

```

```

sdata(pm1).c1HighR = (sdata(pm1).c1profileHighIndR + sdata(pm1).c1rxl -
sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);

sdata(pm1).c1dLowL = (sdata(pm1).c1profileLowIndL + sdata(pm1).c1lxl - sdata(pm1).c1pfxl)*params.cam1Scale(logicP1);
sdata(pm1).c1dMidL = (sdata(pm1).c1profileMidIndL + sdata(pm1).c1lxl - sdata(pm1).c1pfxl)*params.cam1Scale(logicP1);
sdata(pm1).c1dHighL = (sdata(pm1).c1profileHighIndL + sdata(pm1).c1lxl - sdata(pm1).c1pfxl)*params.cam1Scale(logicP1);

sdata(pm1).c1dLowR = (sdata(pm1).c1profileLowIndR + sdata(pm1).c1rxl - sdata(pm1).c1pfxr)*params.cam1Scale(logicP1);
sdata(pm1).c1dMidR = (sdata(pm1).c1profileMidIndR + sdata(pm1).c1rxl - sdata(pm1).c1pfxr)*params.cam1Scale(logicP1);
sdata(pm1).c1dHighR = (sdata(pm1).c1profileHighIndR + sdata(pm1).c1rxl - sdata(pm1).c1pfxr)*params.cam1Scale(logicP1);

sdata(pm1).c1InitialTh = (sdata(pm1).c1pfxr-sdata(pm1).c1pfxl)*params.cam1Scale(logicP1);
sdata(pm1).c1InstantTh = (sdata(pm1).c1MidR-sdata(pm1).c1MidL);
sdata(pm1).c1InitialPos = ((sdata(pm1).c1pfxr+sdata(pm1).c1pfxl)/2-sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
sdata(pm1).c1InstantPos = (sdata(pm1).c1MidR+sdata(pm1).c1MidL)/2;

% parallel calculations
[sdata(pm1).c1kMagLowL,sdata(pm1).c1kArgLowL,sdata(pm1).c1ksxLowL] = ...
    fft1 map(sdata(pm1).c1lineoutLowL,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagMidL,sdata(pm1).c1kArgMidL,sdata(pm1).c1ksxMidL] = ...
    fft1 map(sdata(pm1).c1lineoutMidL,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagHighL,sdata(pm1).c1kArgHighL,sdata(pm1).c1ksxHighL] = ...
    fft1 map(sdata(pm1).c1lineoutHighL,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagTotL,sdata(pm1).c1kArgTotL,sdata(pm1).c1ksxTotL] = ...
    fft1 map(sdata(pm1).c1lineoutTotL,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));

[sdata(pm1).c1kMagLowR,sdata(pm1).c1kArgLowR,sdata(pm1).c1ksxLowR] = ...
    fft1 map(sdata(pm1).c1lineoutLowR,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagMidR,sdata(pm1).c1kArgMidR,sdata(pm1).c1ksxMidR] = ...
    fft1 map(sdata(pm1).c1lineoutMidR,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagHighR,sdata(pm1).c1kArgHighR,sdata(pm1).c1ksxHighR] = ...
    fft1 map(sdata(pm1).c1lineoutHighR,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));
[sdata(pm1).c1kMagTotR,sdata(pm1).c1kArgTotR,sdata(pm1).c1ksxTotR] = ...
    fft1 map(sdata(pm1).c1lineoutTotR,(sdata(pm1).c1lineoutDist(end)-sdata(pm1).c1lineoutDist(1)));

% uncertainties
sdata(pm1).c1dPixUnc = params.cam1Res(logicP1)/params.cam1Scale(logicP1);

sdata(pm1).c1LowUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2)^2+(abs(sdata(pm1).c1LowL)
*params.cam1ScaleUnc(logicP1))^2);
sdata(pm1).c1MidUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2)^2+(abs(sdata(pm1).c1MidL)
*params.cam1ScaleUnc(logicP1))^2);
sdata(pm1).c1HighUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2)^2+(abs(sdata(pm1).c1HighL)*params.
cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1LowUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2)^2+(abs(sdata(pm1).c1LowR)
*params.cam1ScaleUnc(logicP1))^2);

```

```

sdata(pm1).c1MidUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1MidR)
*params.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1HighUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1HighR)*params
.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dLowUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dLowL)
*params.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dMidUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dMidL)
*params.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dHighUncL =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dHighL)*param
s.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dLowUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dLowR)
*params.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dMidUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dMidR)
*params.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dHighUncR =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1))^2)^2+(abs(sdata(pm1).c1dHighR)*param
s.cam1ScaleUnc(logicP1))^2);

sdata(pm1).c1dLowVUncL = (1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dLowUncL)^2+(abs(sdata(pm1).c1dLowL)
*params.laser1Unc(logicP1)/sdata(pm1).cam1Time)^2);

sdata(pm1).c1dMidVUncL = (1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dMidUncL)^2+(abs(sdata(pm1).c1dMidL)
*params.laser1Unc(logicP1)/sdata(pm1).cam1Time)^2);

sdata(pm1).c1dHighVUncL =
(1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dHighUncL)^2+(abs(sdata(pm1).c1dHighL)*params.laser1Unc(logicP1)/sdata(pm1).ca
m1Time)^2);

sdata(pm1).c1dLowVUncR = (1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dLowUncR)^2+(abs(sdata(pm1).c1dLowR)
*params.laser1Unc(logicP1)/sdata(pm1).cam1Time)^2);

sdata(pm1).c1dMidVUncR = (1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dMidUncR)^2+(abs(sdata(pm1).c1dMidR)
*params.laser1Unc(logicP1)/sdata(pm1).cam1Time)^2);

sdata(pm1).c1dHighVUncR =
(1/sdata(pm1).cam1Time)*sqrt((sdata(pm1).c1dHighUncR)^2+(abs(sdata(pm1).c1dHighR)*params.laser1Unc(logicP1)/sdata(pm1).c
am1Time)^2);

sdata(pm1).c1dLowAVuncL = sqrt((sdata(pm1).c1dLowUncL /sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dLowL
^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam1Time^6);

sdata(pm1).c1dMidAVuncL = sqrt((sdata(pm1).c1dMidUncL /sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dMidL
^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam1Time^6);

sdata(pm1).c1dHighAVuncL =
sqrt((sdata(pm1).c1dHighUncL/sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dHighL^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam
1Time^6);

sdata(pm1).c1dLowAVuncR = sqrt((sdata(pm1).c1dLowUncR /sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dLowR
^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam1Time^6);

sdata(pm1).c1dMidAVuncR = sqrt((sdata(pm1).c1dMidUncR /sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dMidR
^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam1Time^6);

sdata(pm1).c1dHighAVuncR =
sqrt((sdata(pm1).c1dHighUncR/sdata(pm1).cam1Time)^2+4*sdata(pm1).c1dHighR^2*params.laser1Unc(logicP1)^2/sdata(pm1).cam
1Time^6);

```

```

sdata(pm1).c1InitialThUnc =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2))^2+(abs(sdata(pm1).c1InitialTh)
*params.cam1ScaleUnc(logicP1))^2);
sdata(pm1).c1InstantThUnc =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2))^2+(abs(sdata(pm1).c1InstantTh)
*params.cam1ScaleUnc(logicP1))^2);
sdata(pm1).c1InitialPosUnc =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2))^2+(abs(sdata(pm1).c1InitialPos)*para
ms.cam1ScaleUnc(logicP1))^2);
sdata(pm1).c1InstantPosUnc =
(1/params.cam1Scale(logicP1))*sqrt(((sdata(pm1).c1dPixUnc*params.cam1Scale(logicP1)^2))^2+(abs(sdata(pm1).c1InstantPos)*par
ams.cam1ScaleUnc(logicP1))^2);

```

```
clear image temp roiL roiR;
```

```
else
```

```

sdata(pm1).c1pxl = NaN;
sdata(pm1).c1pxr = NaN;
sdata(pm1).c1pfxl = NaN;
sdata(pm1).c1pfxr = NaN;
sdata(pm1).c1lyt = NaN;
sdata(pm1).c1lyb = NaN;
sdata(pm1).c1lxl = NaN;
sdata(pm1).c1lxr = NaN;
sdata(pm1).c1rxl = NaN;
sdata(pm1).c1rxr = NaN;

```

```

sdata(pm1).c1profileIntL = NaN;
sdata(pm1).c1profileIntR = NaN;

```

```

sdata(pm1).c1MidDistInd = NaN;
sdata(pm1).c1profileDistL = NaN;
sdata(pm1).c1profileDistR = NaN;
sdata(pm1).c1lineoutDist = NaN;

```

```

sdata(pm1).c1profileLowIntL = NaN;
sdata(pm1).c1profileMidIntL = NaN;
sdata(pm1).c1profileHighIntL = NaN;
sdata(pm1).c1profileLowIndL = NaN;
sdata(pm1).c1profileMidIndL = NaN;
sdata(pm1).c1profileHighIndL = NaN;
sdata(pm1).c1lineoutLowL = NaN;
sdata(pm1).c1lineoutMidL = NaN;
sdata(pm1).c1lineoutHighL = NaN;
sdata(pm1).c1lineoutTotL = NaN;

```

```

sdata(pm1).c1profileLowIntR = NaN;
sdata(pm1).c1profileMidIntR = NaN;
sdata(pm1).c1profileHighIntR = NaN;
sdata(pm1).c1profileLowIndR = NaN;
sdata(pm1).c1profileMidIndR = NaN;

```

sdata(pm1).c1profileHighIndR = NaN;
sdata(pm1).c1lineoutLowR = NaN;
sdata(pm1).c1lineoutMidR = NaN;
sdata(pm1).c1lineoutHighR = NaN;
sdata(pm1).c1lineoutTotR = NaN;

sdata(pm1).c1LowL = NaN;
sdata(pm1).c1MidL = NaN;
sdata(pm1).c1HighL = NaN;

sdata(pm1).c1LowR = NaN;
sdata(pm1).c1MidR = NaN;
sdata(pm1).c1HighR = NaN;

sdata(pm1).c1dLowL = NaN;
sdata(pm1).c1dMidL = NaN;
sdata(pm1).c1dHighL = NaN;

sdata(pm1).c1dLowR = NaN;
sdata(pm1).c1dMidR = NaN;
sdata(pm1).c1dHighR = NaN;

sdata(pm1).c1kMagLowL = NaN;
sdata(pm1).c1kMagMidL = NaN;
sdata(pm1).c1kMagHighL = NaN;
sdata(pm1).c1kMagTotL = NaN;
sdata(pm1).c1kArgLowL = NaN;
sdata(pm1).c1kArgMidL = NaN;
sdata(pm1).c1kArgHighL = NaN;
sdata(pm1).c1kArgTotL = NaN;
sdata(pm1).c1ksxLowL = NaN;
sdata(pm1).c1ksxMidL = NaN;
sdata(pm1).c1ksxHighL = NaN;
sdata(pm1).c1ksxTotL = NaN;

sdata(pm1).c1kMagLowR = NaN;
sdata(pm1).c1kMagMidR = NaN;
sdata(pm1).c1kMagHighR = NaN;
sdata(pm1).c1kMagTotR = NaN;
sdata(pm1).c1kArgLowR = NaN;
sdata(pm1).c1kArgMidR = NaN;
sdata(pm1).c1kArgHighR = NaN;
sdata(pm1).c1kArgTotR = NaN;
sdata(pm1).c1ksxLowR = NaN;
sdata(pm1).c1ksxMidR = NaN;
sdata(pm1).c1ksxHighR = NaN;
sdata(pm1).c1ksxTotR = NaN;

sdata(pm1).c1InitialTh = NaN;

```

sdata(pm1).c1InstantTh = NaN;
sdata(pm1).c1InitialPos = NaN;
sdata(pm1).c1InstantPos = NaN;

sdata(pm1).c1dPixUnc = NaN;

sdata(pm1).c1LowUncL = NaN;
sdata(pm1).c1MidUncL = NaN;
sdata(pm1).c1HighUncL = NaN;

sdata(pm1).c1LowUncR = NaN;
sdata(pm1).c1MidUncR = NaN;
sdata(pm1).c1HighUncR = NaN;

sdata(pm1).c1dLowUncL = NaN;
sdata(pm1).c1dMidUncL = NaN;
sdata(pm1).c1dHighUncL = NaN;

sdata(pm1).c1dLowUncR = NaN;
sdata(pm1).c1dMidUncR = NaN;
sdata(pm1).c1dHighUncR = NaN;

sdata(pm1).c1dLowVUncL = NaN;
sdata(pm1).c1dMidVUncL = NaN;
sdata(pm1).c1dHighVUncL = NaN;

sdata(pm1).c1dLowVUncR = NaN;
sdata(pm1).c1dMidVUncR = NaN;
sdata(pm1).c1dHighVUncR = NaN;

sdata(pm1).c1dLowAUncL = NaN;
sdata(pm1).c1dMidAUncL = NaN;
sdata(pm1).c1dHighAUncL = NaN;

sdata(pm1).c1dLowAUncR = NaN;
sdata(pm1).c1dMidAUncR = NaN;
sdata(pm1).c1dHighAUncR = NaN;

sdata(pm1).c1InitialThUnc = NaN;
sdata(pm1).c1InstantThUnc = NaN;
sdata(pm1).c1InitialPosUnc = NaN;
sdata(pm1).c1InstantPosUnc = NaN;

end

% Camera 2
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0
    logic2 = imap(:,3)==params.cam2ID(logicP1);

```

```

logic3 = imap(:,4)==params.cam2PreID(logicP1);
logic4 = imap(:,4)==params.cam2ImID(logicP1);
timap = imap(logic1 & logic2 & logic4,:);

[image,temp] = load_images(timap(1,2:4),imType,shotPath);

sdata(pm1).c2pxl = params.cam2pxl(logicP1);
sdata(pm1).c2pxr = sdata(pm1).c2pxl + params.cam2pxld(logicP1);
sdata(pm1).c2pfxl = params.cam2pfl(logicP1);
sdata(pm1).c2pfxr = sdata(pm1).c2pfxl + params.cam2pfld(logicP1);
sdata(pm1).c2yt = params.cam2yt(logicP1);
sdata(pm1).c2yb = params.cam2yt(logicP1)+params.cam2yd(logicP1);
sdata(pm1).c2lxl = params.cam2x1(logicP1);
sdata(pm1).c2lxr = params.cam2x2(logicP1);
sdata(pm1).c2rxl = params.cam2x3(logicP1);
sdata(pm1).c2rxr = params.cam2x4(logicP1);

roiL = image.data(sdata(pm1).c2yt:sdata(pm1).c2yb,sdata(pm1).c2lxl:sdata(pm1).c2lxr,2);
roiR = image.data(sdata(pm1).c2yt:sdata(pm1).c2yb,sdata(pm1).c2rxl:sdata(pm1).c2rxr,2);

sdata(pm1).c2profileIntL = mean(roiL,1);
sdata(pm1).c2profileIntR = mean(roiR,1);

sdata(pm1).c2MidDistInd = round((sdata(pm1).c2pxr + sdata(pm1).c2pxl)/2);
sdata(pm1).c2profileDistL = ((sdata(pm1).c2lxl:sdata(pm1).c2lxr)-sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2profileDistR = ((sdata(pm1).c2rxl:sdata(pm1).c2rxr)-sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2lineoutDist = ((sdata(pm1).c2yt:sdata(pm1).c2yb) -sdata(pm1).c2yt)*params.cam2Scale(logicP1);

sdata(pm1).c2profileLowIntL = (max(sdata(pm1).c2profileIntL)-min(sdata(pm1).c2profileIntL))*profileLow
+min(sdata(pm1).c2profileIntL);
sdata(pm1).c2profileMidIntL = mean(sdata(pm1).c2profileIntL);
sdata(pm1).c2profileHighIntL = (max(sdata(pm1).c2profileIntL)-
min(sdata(pm1).c2profileIntL))*profileHigh+min(sdata(pm1).c2profileIntL);
sdata(pm1).c2profileLowIndL = find(sdata(pm1).c2profileIntL >= sdata(pm1).c2profileLowIntL ,1,'last');
sdata(pm1).c2profileMidIndL1 = find(sdata(pm1).c2profileIntL >= sdata(pm1).c2profileMidIntL ,1,'last');
sdata(pm1).c2profileMidIndL2 = find(sdata(pm1).c2profileIntL <= sdata(pm1).c2profileMidIntL ,1,'first');
sdata(pm1).c2profileMidIndL = round(mean([sdata(pm1).c2profileMidIndL1,sdata(pm1).c2profileMidIndL2]));
sdata(pm1).c2profileHighIndL = find(sdata(pm1).c2profileIntL <= sdata(pm1).c2profileHighIntL,1,'first');

% sdata(pm1).c2profileMidIndL = round((sdata(pm1).c2profileHighIndL-
sdata(pm1).c2profileLowIndL)/2+sdata(pm1).c2profileLowIndL);
sdata(pm1).c2profileLowIntL = sdata(pm1).c2profileIntL(sdata(pm1).c2profileLowIndL);
sdata(pm1).c2profileMidIntL = sdata(pm1).c2profileIntL(sdata(pm1).c2profileMidIndL);
sdata(pm1).c2profileHighIntL = sdata(pm1).c2profileIntL(sdata(pm1).c2profileHighIndL);

sdata(pm1).c2lineoutLowL = 255-mean(roiL(:,sdata(pm1).c2profileLowIndL -lineoutAvr:sdata(pm1).c2profileLowIndL
+lineoutAvr),2);
sdata(pm1).c2lineoutMidL = 255-mean(roiL(:,sdata(pm1).c2profileMidIndL -lineoutAvr:sdata(pm1).c2profileMidIndL
+lineoutAvr),2);
sdata(pm1).c2lineoutHighL = 255-mean(roiL(:,sdata(pm1).c2profileHighIndL-
lineoutAvr:sdata(pm1).c2profileHighIndL+lineoutAvr),2);

```

```

sdata(pm1).c2lineoutTotL = 255-mean(roiL(:,sdata(pm1).c2profileHighIndL:sdata(pm1).c2profileLowIndL),2);

sdata(pm1).c2profileLowIntR = (max(sdata(pm1).c2profileIntR)-min(sdata(pm1).c2profileIntR))*profileLow
+min(sdata(pm1).c2profileIntR);
sdata(pm1).c2profileMidIntR = mean(sdata(pm1).c2profileIntR);
sdata(pm1).c2profileHighIntR = (max(sdata(pm1).c2profileIntR)-
min(sdata(pm1).c2profileIntR))*profileHigh+min(sdata(pm1).c2profileIntR);
sdata(pm1).c2profileLowIndR = find(sdata(pm1).c2profileIntR >= sdata(pm1).c2profileLowIntR ,1,'first');
sdata(pm1).c2profileMidIndR1 = find(sdata(pm1).c2profileIntR >= sdata(pm1).c2profileMidIntR ,1,'first');
sdata(pm1).c2profileMidIndR2 = find(sdata(pm1).c2profileIntR <= sdata(pm1).c2profileMidIntR ,1,'last');
sdata(pm1).c2profileMidIndR = round(mean([sdata(pm1).c2profileMidIndR1,sdata(pm1).c2profileMidIndR2]));
sdata(pm1).c2profileHighIndR = find(sdata(pm1).c2profileIntR <= sdata(pm1).c2profileHighIntR,1,'last');

% sdata(pm1).c2profileMidIndR = round((sdata(pm1).c2profileHighIndR-
sdata(pm1).c2profileLowIndR)/2+sdata(pm1).c2profileLowIndR);
sdata(pm1).c2profileLowIntR = sdata(pm1).c2profileIntR(sdata(pm1).c2profileLowIndR);
sdata(pm1).c2profileMidIntR = sdata(pm1).c2profileIntR(sdata(pm1).c2profileMidIndR);
sdata(pm1).c2profileHighIntR = sdata(pm1).c2profileIntR(sdata(pm1).c2profileHighIndR);

sdata(pm1).c2lineoutLowR = 255-mean(roiR(:,sdata(pm1).c2profileLowIndR -lineoutAvr:sdata(pm1).c2profileLowIndR
+lineoutAvr),2);
sdata(pm1).c2lineoutMidR = 255-mean(roiR(:,sdata(pm1).c2profileMidIndR -lineoutAvr:sdata(pm1).c2profileMidIndR
+lineoutAvr),2);
sdata(pm1).c2lineoutHighR = 255-mean(roiR(:,sdata(pm1).c2profileHighIndR-
lineoutAvr:sdata(pm1).c2profileHighIndR+lineoutAvr),2);
sdata(pm1).c2lineoutTotR = 255-mean(roiR(:,sdata(pm1).c2profileLowIndR:sdata(pm1).c2profileHighIndR),2);

% perpendicular calculaions
sdata(pm1).c2LowL = (sdata(pm1).c2profileLowIndL + sdata(pm1).c2lx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2MidL = (sdata(pm1).c2profileMidIndL + sdata(pm1).c2lx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2HighL = (sdata(pm1).c2profileHighIndL + sdata(pm1).c2lx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);

sdata(pm1).c2LowR = (sdata(pm1).c2profileLowIndR + sdata(pm1).c2rx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2MidR = (sdata(pm1).c2profileMidIndR + sdata(pm1).c2rx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
sdata(pm1).c2HighR = (sdata(pm1).c2profileHighIndR + sdata(pm1).c2rx1 -
sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);

sdata(pm1).c2dLowL = (sdata(pm1).c2profileLowIndL + sdata(pm1).c2lx1 - sdata(pm1).c2pfxl)*params.cam2Scale(logicP1);
sdata(pm1).c2dMidL = (sdata(pm1).c2profileMidIndL + sdata(pm1).c2lx1 - sdata(pm1).c2pfxl)*params.cam2Scale(logicP1);
sdata(pm1).c2dHighL = (sdata(pm1).c2profileHighIndL + sdata(pm1).c2lx1 - sdata(pm1).c2pfxl)*params.cam2Scale(logicP1);

sdata(pm1).c2dLowR = (sdata(pm1).c2profileLowIndR + sdata(pm1).c2rx1 - sdata(pm1).c2pfxr)*params.cam2Scale(logicP1);
sdata(pm1).c2dMidR = (sdata(pm1).c2profileMidIndR + sdata(pm1).c2rx1 - sdata(pm1).c2pfxr)*params.cam2Scale(logicP1);
sdata(pm1).c2dHighR = (sdata(pm1).c2profileHighIndR + sdata(pm1).c2rx1 - sdata(pm1).c2pfxr)*params.cam2Scale(logicP1);

sdata(pm1).c2InitialTh = (sdata(pm1).c2pfxr-sdata(pm1).c2pfxl)*params.cam2Scale(logicP1);
sdata(pm1).c2InstantTh = (sdata(pm1).c2MidR-sdata(pm1).c2MidL);
sdata(pm1).c2InitialPos = ((sdata(pm1).c2pfxr+sdata(pm1).c2pfxl)/2-sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);

```

```

sdata(pm1).c2InstantPos = (sdata(pm1).c2MidR+sdata(pm1).c2MidL)/2;

% parallel calculations
[sdata(pm1).c2kMagLowL,sdata(pm1).c2kArgLowL,sdata(pm1).c2ksxLowL] = ...
    fft1 map(sdata(pm1).c2lineoutLowL,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagMidL,sdata(pm1).c2kArgMidL,sdata(pm1).c2ksxMidL] = ...
    fft1 map(sdata(pm1).c2lineoutMidL,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagHighL,sdata(pm1).c2kArgHighL,sdata(pm1).c2ksxHighL] = ...
    fft1 map(sdata(pm1).c2lineoutHighL,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagTotL,sdata(pm1).c2kArgTotL,sdata(pm1).c2ksxTotL] = ...
    fft1 map(sdata(pm1).c2lineoutTotL,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));

[sdata(pm1).c2kMagLowR,sdata(pm1).c2kArgLowR,sdata(pm1).c2ksxLowR] = ...
    fft1 map(sdata(pm1).c2lineoutLowR,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagMidR,sdata(pm1).c2kArgMidR,sdata(pm1).c2ksxMidR] = ...
    fft1 map(sdata(pm1).c2lineoutMidR,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagHighR,sdata(pm1).c2kArgHighR,sdata(pm1).c2ksxHighR] = ...
    fft1 map(sdata(pm1).c2lineoutHighR,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));
[sdata(pm1).c2kMagTotR,sdata(pm1).c2kArgTotR,sdata(pm1).c2ksxTotR] = ...
    fft1 map(sdata(pm1).c2lineoutTotR,(sdata(pm1).c2lineoutDist(end)-sdata(pm1).c2lineoutDist(1)));

% uncertainties
sdata(pm1).c2dPixUnc = params.cam2Res(logicP1)/params.cam2Scale(logicP1);

sdata(pm1).c2LowUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2LowL)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2MidUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2MidL)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2HighUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2HighL)*params.
cam2ScaleUnc(logicP1))^2);

sdata(pm1).c2LowUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2LowR)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2MidUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2MidR)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2HighUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2HighR)*params.
cam2ScaleUnc(logicP1))^2);

sdata(pm1).c2dLowUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2dLowL)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2dMidUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2dMidL)
*params.cam2ScaleUnc(logicP1))^2);
sdata(pm1).c2dHighUncL =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2)^2+(abs(sdata(pm1).c2dHighL)*param
s.cam2ScaleUnc(logicP1))^2);

```

```

    sdata(pm1).c2dLowUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2dLowR)
*params.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2dMidUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2dMidR)
*params.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2dHighUncR =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2dHighR)*param
s.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2dLowVUncL = (1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dLowUncL)^2+(abs(sdata(pm1).c2dLowL)
*params.laser2Unc(logicP1)/sdata(pm1).cam2Time)^2);

    sdata(pm1).c2dMidVUncL = (1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dMidUncL)^2+(abs(sdata(pm1).c2dMidL)
*params.laser2Unc(logicP1)/sdata(pm1).cam2Time)^2);

    sdata(pm1).c2dHighVUncL =
(1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dHighUncL)^2+(abs(sdata(pm1).c2dHighL)*params.laser2Unc(logicP1)/sdata(pm1).ca
m2Time)^2);

    sdata(pm1).c2dLowVUncR = (1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dLowUncR)^2+(abs(sdata(pm1).c2dLowR)
*params.laser2Unc(logicP1)/sdata(pm1).cam2Time)^2);

    sdata(pm1).c2dMidVUncR = (1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dMidUncR)^2+(abs(sdata(pm1).c2dMidR)
*params.laser2Unc(logicP1)/sdata(pm1).cam2Time)^2);

    sdata(pm1).c2dHighVUncR =
(1/sdata(pm1).cam2Time)*sqrt((sdata(pm1).c2dHighUncR)^2+(abs(sdata(pm1).c2dHighR)*params.laser2Unc(logicP1)/sdata(pm1).c
am2Time)^2);

    sdata(pm1).c2dLowAVuncL = sqrt((sdata(pm1).c2dLowUncL /sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dLowL
^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam2Time^6);

    sdata(pm1).c2dMidAVuncL = sqrt((sdata(pm1).c2dMidUncL /sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dMidL
^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam2Time^6);

    sdata(pm1).c2dHighAVuncL =
sqrt((sdata(pm1).c2dHighUncL/sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dHighL^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam
2Time^6);

    sdata(pm1).c2dLowAVuncR = sqrt((sdata(pm1).c2dLowUncR /sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dLowR
^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam2Time^6);

    sdata(pm1).c2dMidAVuncR = sqrt((sdata(pm1).c2dMidUncR /sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dMidR
^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam2Time^6);

    sdata(pm1).c2dHighAVuncR =
sqrt((sdata(pm1).c2dHighUncR/sdata(pm1).cam2Time)^2+4*sdata(pm1).c2dHighR^2*params.laser2Unc(logicP1)^2/sdata(pm1).cam
2Time^6);

    sdata(pm1).c2InitialThUnc =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2InitialTh)
*params.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2InstantThUnc =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2InstantTh)
*params.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2InitialPosUnc =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2InitialPos)*para
ms.cam2ScaleUnc(logicP1))^2);

    sdata(pm1).c2InstantPosUnc =
(1/params.cam2Scale(logicP1))*sqrt(((sdata(pm1).c2dPixUnc*params.cam2Scale(logicP1)^2))^2+(abs(sdata(pm1).c2InstantPos)*par
ams.cam2ScaleUnc(logicP1))^2);

```

```
clear image temp roiL roiR;
```

```
else
```

sdata(pm1).c2pxl = NaN;
sdata(pm1).c2pxr = NaN;
sdata(pm1).c2pfxl = NaN;
sdata(pm1).c2pfxr = NaN;
sdata(pm1).c2yt = NaN;
sdata(pm1).c2yb = NaN;
sdata(pm1).c2lxl = NaN;
sdata(pm1).c2lxr = NaN;
sdata(pm1).c2rxl = NaN;
sdata(pm1).c2rxr = NaN;

sdata(pm1).c2profileIntL = NaN;
sdata(pm1).c2profileIntR = NaN;

sdata(pm1).c2MidDistInd = NaN;
sdata(pm1).c2profileDistL = NaN;
sdata(pm1).c2profileDistR = NaN;
sdata(pm1).c2lineoutDist = NaN;

sdata(pm1).c2profileLowIntL = NaN;
sdata(pm1).c2profileMidIntL = NaN;
sdata(pm1).c2profileHighIntL = NaN;
sdata(pm1).c2profileLowIndL = NaN;
sdata(pm1).c2profileMidIndL = NaN;
sdata(pm1).c2profileHighIndL = NaN;
sdata(pm1).c2lineoutLowL = NaN;
sdata(pm1).c2lineoutMidL = NaN;
sdata(pm1).c2lineoutHighL = NaN;
sdata(pm1).c2lineoutTotL = NaN;

sdata(pm1).c2profileLowIntR = NaN;
sdata(pm1).c2profileMidIntR = NaN;
sdata(pm1).c2profileHighIntR = NaN;
sdata(pm1).c2profileLowIndR = NaN;
sdata(pm1).c2profileMidIndR = NaN;
sdata(pm1).c2profileHighIndR = NaN;
sdata(pm1).c2lineoutLowR = NaN;
sdata(pm1).c2lineoutMidR = NaN;
sdata(pm1).c2lineoutHighR = NaN;
sdata(pm1).c2lineoutTotR = NaN;

sdata(pm1).c2LowL = NaN;
sdata(pm1).c2MidL = NaN;
sdata(pm1).c2HighL = NaN;

sdata(pm1).c2LowR = NaN;
sdata(pm1).c2MidR = NaN;
sdata(pm1).c2HighR = NaN;

```
sdata(pm1).c2dLowL = NaN;
sdata(pm1).c2dMidL = NaN;
sdata(pm1).c2dHighL = NaN;

sdata(pm1).c2dLowUncL = NaN;
sdata(pm1).c2dMidUncL = NaN;
sdata(pm1).c2dHighUncL = NaN;

sdata(pm1).c2dLowR = NaN;
sdata(pm1).c2dMidR = NaN;
sdata(pm1).c2dHighR = NaN;

sdata(pm1).c2dLowUncR = NaN;
sdata(pm1).c2dMidUncR = NaN;
sdata(pm1).c2dHighUncR = NaN;

sdata(pm1).c2kMagLowL = NaN;
sdata(pm1).c2kMagMidL = NaN;
sdata(pm1).c2kMagHighL = NaN;
sdata(pm1).c2kMagTotL = NaN;
sdata(pm1).c2kArgLowL = NaN;
sdata(pm1).c2kArgMidL = NaN;
sdata(pm1).c2kArgHighL = NaN;
sdata(pm1).c2kArgTotL = NaN;
sdata(pm1).c2ksxLowL = NaN;
sdata(pm1).c2ksxMidL = NaN;
sdata(pm1).c2ksxHighL = NaN;
sdata(pm1).c2ksxTotL = NaN;

sdata(pm1).c2kMagLowR = NaN;
sdata(pm1).c2kMagMidR = NaN;
sdata(pm1).c2kMagHighR = NaN;
sdata(pm1).c2kMagTotR = NaN;
sdata(pm1).c2kArgLowR = NaN;
sdata(pm1).c2kArgMidR = NaN;
sdata(pm1).c2kArgHighR = NaN;
sdata(pm1).c2kArgTotR = NaN;
sdata(pm1).c2ksxLowR = NaN;
sdata(pm1).c2ksxMidR = NaN;
sdata(pm1).c2ksxHighR = NaN;
sdata(pm1).c2ksxTotR = NaN;

sdata(pm1).c2InitialTh = NaN;
sdata(pm1).c2InstantTh = NaN;
sdata(pm1).c2InitialPos = NaN;
sdata(pm1).c2InstantPos = NaN;

sdata(pm1).c2dPixUnc = NaN;
```

```

sdata(pm1).c2LowUncL = NaN;
sdata(pm1).c2MidUncL = NaN;
sdata(pm1).c2HighUncL = NaN;

sdata(pm1).c2LowUncR = NaN;
sdata(pm1).c2MidUncR = NaN;
sdata(pm1).c2HighUncR = NaN;

sdata(pm1).c2dLowUncL = NaN;
sdata(pm1).c2dMidUncL = NaN;
sdata(pm1).c2dHighUncL = NaN;

sdata(pm1).c2dLowUncR = NaN;
sdata(pm1).c2dMidUncR = NaN;
sdata(pm1).c2dHighUncR = NaN;

sdata(pm1).c2dLowVUncL = NaN;
sdata(pm1).c2dMidVUncL = NaN;
sdata(pm1).c2dHighVUncL = NaN;

sdata(pm1).c2dLowVUncR = NaN;
sdata(pm1).c2dMidVUncR = NaN;
sdata(pm1).c2dHighVUncR = NaN;

sdata(pm1).c2dLowAUncL = NaN;
sdata(pm1).c2dMidAUncL = NaN;
sdata(pm1).c2dHighAUncL = NaN;

sdata(pm1).c2dLowAUncR = NaN;
sdata(pm1).c2dMidAUncR = NaN;
sdata(pm1).c2dHighAUncR = NaN;

sdata(pm1).c2InitialThUnc = NaN;
sdata(pm1).c2InstantThUnc = NaN;
sdata(pm1).c2InitialPosUnc = NaN;
sdata(pm1).c2InstantPosUnc = NaN;

end

end

toc
end
% ===== end =====

% ===== start =====
% Plot Perpendicular Profiles
if genPerPro

```

```

display('Plotting Perpendicular Profiles');

for k=1:snum
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    % Camera 1 Left
    if genPerProC1L
        logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
        if sum(logicP1) > 0

            hf1=figure('Visible','off');

            a1p = [0,0,1,1];
            a2p = [0,0,1,1];

            set(gcf,'paperunits','inches');
            set(gcf,'paperposition',[0,0,figOutX,figOutY]);
            set(gcf,'paperpositionmode','auto');
            set(gcf,'PaperSize',[figOutX,figOutY]);
            set(gcf,'units','inches');
            set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
            set(gcf,'units','normalized');

            ha1 = axes(...
                'Position',a1p,...
                'Visible','off...
            );

            ha2 = axes(...
                'Position',a2p,...
                'activepositionproperty','position',...
                'XMinorTick','on',...
                'YMinorTick','on',...
                'FontSize',fontSize,...
                'color','w'...
            );

            set(hf1,'CurrentAxes',ha2);

            lgdnames = { 'Transmission',[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f')
            '% Threshold'],'Initial Left','Initial Right'};
            lgdlogic = true(length(lgdnames),1);
            lgdhandles = [];
            h11=line( sdata(pm1).c1profileDistL*1e3, ...
                sdata(pm1).c1profileIntL, ...
                'color','k','linewidth',2.0);
            lgdhandles = [lgdhandles,h11];
            h12=line( sdata(pm1).c1profileDistL(sdata(pm1).c1profileLowIndL)*1e3, ...

```

```

sdata(pm1).c1profileLowIntL, ...
'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','r');
lgdhandles = [lgdhandles,h12];
hl3=line( sdata(pm1).c1profileDistL(sdata(pm1).c1profileMidIndL)*1e3, ...
sdata(pm1).c1profileMidIntL, ...
'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
lgdhandles = [lgdhandles,h13];
hl4=line( sdata(pm1).c1profileDistL(sdata(pm1).c1profileHighIndL)*1e3, ...
sdata(pm1).c1profileHighIntL, ...
'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','b');
lgdhandles = [lgdhandles,h14];

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

cenDistL = (sdata(pm1).c1pfxl - sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
if (cenDistL >= sdata(pm1).c1profileDistL(1) && cenDistL <= sdata(pm1).c1profileDistL(end))
    hl5=line([cenDistL,cenDistL]*1e3,[fytick(1),fytick(end)], ...
'linestyle','-','linewidth',1.5,'color',darkSlateGrey);
    lgdhandles = [lgdhandles,h15];
else
    lgdlogic(5)=false;
end

cenDistR = (sdata(pm1).c1pfxr - sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
if (cenDistR >= sdata(pm1).c1profileDistL(1) && cenDistR <= sdata(pm1).c1profileDistL(end))
    hl6=line([cenDistR,cenDistR]*1e3,[fytick(1),fytick(end)], ...
'linestyle','-','linewidth',1.5,'color',lightSlateGrey);
    lgdhandles = [lgdhandles,h16];
else
    lgdlogic(6)=false;
end

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontSize',fontSize);
ylabel('\bfLaser Transmission (AU)\rm','fontSize',fontSize);

title(['\bfPerpendicular Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
```

```

fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j), '%3.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 1',...
    'Image = 2',...
    'Side = Left',...
    },...
    ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam1Cur),'%4.0f') ' kA']...
    ['Initial Left = ' num2str(cenDistL*1e3,'%2.1f') ' mm']...
    ['Initial Right = ' num2str(cenDistR*1e3,'%2.1f') ' mm']...
    ]);

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...

```

```

        );
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-001_Image-002_Side-L'];

print(hf1, '-d.tif', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'perpProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'perpProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 1 Right
if genPerProC1R
    logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
    if sum(logicP1) > 0

```

```

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = { 'Transmission',[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f')
'% Threshold'],'Initial Left','Initial Right'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
h1=line( sdata(pm1).c1profileDistR*1e3, ...
    sdata(pm1).c1profileIntR, ...
    'color','k','linewidth',2.0);
lgdhandles = [lgdhandles,h1];
h2=line( sdata(pm1).c1profileDistR(sdata(pm1).c1profileLowIndR)*1e3, ...
    sdata(pm1).c1profileLowIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','r');
lgdhandles = [lgdhandles,h2];
h3=line( sdata(pm1).c1profileDistR(sdata(pm1).c1profileMidIndR)*1e3, ...
    sdata(pm1).c1profileMidIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
lgdhandles = [lgdhandles,h3];
h4=line( sdata(pm1).c1profileDistR(sdata(pm1).c1profileHighIndR)*1e3, ...
    sdata(pm1).c1profileHighIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','b');
lgdhandles = [lgdhandles,h4];

```

```

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

cenDistL = (sdata(pm1).c1pfxl - sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
if (cenDistL >= sdata(pm1).c1profileDistR(1) && cenDistL <= sdata(pm1).c1profileDistR(end))
    hl5=line([cenDistL,cenDistL]*1e3,[fytick(1),fytick(end)], ...
            'linestyle','-', 'linewidth',1.5, 'color',darkSlateGrey);
    lgdhandles = [lgdhandles,hl5];
else
    lgdlogic(5)=false;
end

cenDistR = (sdata(pm1).c1pfxr - sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1);
if (cenDistR >= sdata(pm1).c1profileDistR(1) && cenDistR <= sdata(pm1).c1profileDistR(end))
    hl6=line([cenDistR,cenDistR]*1e3,[fytick(1),fytick(end)], ...
            'linestyle','-', 'linewidth',1.5, 'color',lightSlateGrey);
    lgdhandles = [lgdhandles,hl6];
else
    lgdlogic(6)=false;
end

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm', 'fontSize',fontSize);
ylabel('\bfLaser Transmission (AU)\rm', 'fontSize',fontSize);

title(['\bfPerpendicular Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'], 'fontSize',fontSize);

grid on;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic), 'fontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

```

```

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    {...
    '\bfShot Parameters\rm',...
    'Camera = 1',...
    'Image = 2',...
    'Side = Right',...
    ",...
    ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam1Cur),'%4.0f') ' kA']...
    ['Initial Left = ' num2str(cenDistL*1e3,'%2.1f') ' mm']...
    ['Initial Right = ' num2str(cenDistR*1e3,'%2.1f') ' mm']...
    });

```

```

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

```

```

tp = get(ht1,'position');

```

```

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

```

```

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )

```

```

    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);

```

```

    a2p = get(ha2,'position');

```

```

else

```

```

    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);

```

```

    a2p = get(ha2,'position');

```

```

end

```

```

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);

```

```

lgdp = get(hlgd1,'position');

```

```

set(ht1,'position',[...

```

```

lgdp(1),...
lgdp(2)-txtOff/figOutY-tp(4),...
tp(3),...
tp(4));

filename = [ ...
'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
'_Camera-001_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'perpProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'perpProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 2 Left
if genPerProc2L
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off...
    );

```

```

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);

lgnames = { 'Transmission',[num2str(profileLow*100,'%3.0f') '% Threshold'], 'Mean Position',[num2str(profileHigh*100,'%3.0f')
'% Threshold'],'Initial Left','Initial Right'};
lgdlogic = true(length(lgnames),1);
lgdhandles = [];
hl1=line( sdata(pm1).c2profileDistL*1e3, ...
    sdata(pm1).c2profileIntL, ...
    'color','k','linewidth',2.0);
lgdhandles = [lgdhandles,hl1];
hl2=line( sdata(pm1).c2profileDistL(sdata(pm1).c2profileLowIndL)*1e3, ...
    sdata(pm1).c2profileLowIntL, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','r');
lgdhandles = [lgdhandles,hl2];
hl3=line( sdata(pm1).c2profileDistL(sdata(pm1).c2profileMidIndL)*1e3, ...
    sdata(pm1).c2profileMidIntL, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
lgdhandles = [lgdhandles,hl3];
hl4=line( sdata(pm1).c2profileDistL(sdata(pm1).c2profileHighIndL)*1e3, ...
    sdata(pm1).c2profileHighIntL, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','b');
lgdhandles = [lgdhandles,hl4];

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

cenDistL = (sdata(pm1).c2pfxl - sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
if (cenDistL >= sdata(pm1).c2profileDistL(1) && cenDistL <= sdata(pm1).c2profileDistL(end))
    hl5=line([cenDistL,cenDistL]*1e3,[fytick(1),fytick(end)], ...
        'linestyle','-','linewidth',1.5,'color',darkSlateGrey);
    lgdhandles = [lgdhandles,hl5];
else
    lgdlogic(5)=false;
end

cenDistR = (sdata(pm1).c2pfxr - sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);

```

```

if (cenDistR >= sdata(pm1).c2profileDistL(1) && cenDistR <= sdata(pm1).c2profileDistL(end))
    hl6=line([cenDistR,cenDistR]*1e3,[fytick(1),fytick(end)], ...
        'linestyle','--','linewidth',1.5,'color',lightSlateGrey);
    lgdhandles = [lgdhandles,hl6];
else
    lgdlogic(6)=false;
end

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontsize',fontSize);
ylabel('\bfLaser Transmission (AU)\rm','fontsize',fontSize);

title(['\bfPerpendicular Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontsize',fontSize);

grid on;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontsize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 2',...
    'Image = 2',...
    'Side = Left',...
    },...
    ['Time = ' num2str(sdata(k).cam2Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam2Cur),'%4.0f') ' kA']...
    ['Initial Left = ' num2str(cenDistL*1e3,'%2.1f') ' mm']...
    ['Initial Right = ' num2str(cenDistR*1e3,'%2.1f') ' mm']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

```

```

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY -tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-L'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\ '], 'perpProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'perpProfiles']);
else
    display( mess );

```

```

        display( num2str(pmap(k,2)) );
        break;
    end

    close(hf1);
end

end

% Camera 2 Right
if genPerProC2R
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off'...
    );

    ha2 = axes(...
        'Position',a2p,...
        'activepositionproperty','position',...
        'XMinorTick','on',...
        'YMinorTick','on',...
        'FontSize',fontSize,...
        'color','w'...
    );

    set(hf1,'CurrentAxes',ha2);
    lgdnames = { Transmission',[num2str(profileLow*100,'%3.0f') '% Threshold'], Mean Position',[num2str(profileHigh*100,'%3.0f')
'% Threshold'], Initial Left, Initial Right};
    lgdlogic = true(length(lgdnames),1);
    lgdhandles = [];
    hl1=line( sdata(pm1).c2profileDistR*1e3, ...
        sdata(pm1).c2profileIntR, ...
        'color','k','linewidth',2.0);

```

```

    lgdhandles = [lgdhandles,h11];
h12=line( sdata(pm1).c2profileDistR(sdata(pm1).c2profileLowIndR)*1e3, ...
    sdata(pm1).c2profileLowIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','r');
    lgdhandles = [lgdhandles,h12];
h13=line( sdata(pm1).c2profileDistR(sdata(pm1).c2profileMidIndR)*1e3, ...
    sdata(pm1).c2profileMidIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,h13];
h14=line( sdata(pm1).c2profileDistR(sdata(pm1).c2profileHighIndR)*1e3, ...
    sdata(pm1).c2profileHighIntR, ...
    'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','b');
    lgdhandles = [lgdhandles,h14];

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

cenDistL = (sdata(pm1).c2pfxl - sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
if (cenDistL >= sdata(pm1).c2profileDistR(1) && cenDistL <= sdata(pm1).c2profileDistR(end))
    h15=line([cenDistL,cenDistL]*1e3,[fytick(1),fytick(end)], ...
        'linestyle','-','linewidth',1.5,'color',darkSlateGrey);
    lgdhandles = [lgdhandles,h15];
else
    lgdlogic(5)=false;
end

cenDistR = (sdata(pm1).c2pfxr - sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1);
if (cenDistR >= sdata(pm1).c2profileDistR(1) && cenDistR <= sdata(pm1).c2profileDistR(end))
    h16=line([cenDistR,cenDistR]*1e3,[fytick(1),fytick(end)], ...
        'linestyle','-','linewidth',1.5,'color',lightSlateGrey);
    lgdhandles = [lgdhandles,h16];
else
    lgdlogic(6)=false;
end

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontsize',fontSize);
ylabel('\bfLaser Transmission (AU)\rm','fontsize',fontSize);

title(['\bfPerpendicular Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontsize',fontSize);

grid on;
box on;

```

```

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 2',...
    'Image = 2',...
    'Side = Right',...
    },...
    ['Time = ' num2str(sdata(k).cam2Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam2Cur),'%4.0f') ' kA']...
    ['Initial Left = ' num2str(cenDistL*1e3,'%2.1f') ' mm']...
    ['Initial Right = ' num2str(cenDistR*1e3,'%2.1f') ' mm']...
    ]);

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...

```

```

        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ];
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgd(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'perpProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\perpProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

end

toc
end
% ===== end =====

% ===== start =====
% Plot Parallel Profiles

```

```

if genParPro
display('Plotting Parallel Profiles');

for k=1:snum
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    % Camera 1 Left
    if genParProC1L
        logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
        if sum(logicP1) > 0

            hf1=figure('Visible','off');

            a1p = [0,0,1,1];
            a2p = [0,0,1,1];

            set(gcf,'paperunits','inches');
            set(gcf,'paperposition',[0,0,figOutX,figOutY]);
            set(gcf,'paperpositionmode','auto');
            set(gcf,'PaperSize', [figOutX,figOutY]);
            set(gcf,'units','inches');
            set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
            set(gcf,'units','normalized');

            ha1 = axes(...
                'Position',a1p,...
                'Visible','off'...
            );

            ha2 = axes(...
                'Position',a2p,...
                'activepositionproperty','position',...
                'XMinorTick','on',...
                'YMinorTick','on',...
                'FontSize',fontSize,...
                'color','w'...
            );

            set(hf1,'CurrentAxes',ha2);

            lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'], 'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
            Threshold'],'Average'};
            lgdlogic = true(length(lgdnames),1);
            lgdhandles = [];
            hl1=line( sdata(pm1).c1lineoutDist*1e3, ...
                sdata(pm1).c1lineoutLowL, ...
                'color','r','linewidth',1.0,'linestyle','--');
            lgdhandles = [lgdhandles,hl1];

```

```

hl2=line( sdata(pm1).c1lineoutDist*1e3, ...
    sdata(pm1).c1lineoutMidL, ...
    'color','g','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl2];
hl3=line( sdata(pm1).c1lineoutDist*1e3, ...
    sdata(pm1).c1lineoutHighL, ...
    'color','b','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl3];
hl4=line( sdata(pm1).c1lineoutDist*1e3, ...
    sdata(pm1).c1lineoutTotL, ...
    'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontSize',fontSize);
ylabel('\bfInverted Laser Transmission (AU)\rm','fontSize',fontSize);

title(['\bfParallel Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 1',...

```

```

    'Image = 2',...
    'Side = Left',...
    ",...
    ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f) ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam1Cur,'%4.0f) ' kA']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY -tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f) ...
    '_Camera-001_Image-002_Side-L'];

```

```

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'parProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\parProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 1 Right
if genParProC1R
logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off'...
    );

    ha2 = axes(...
        'Position',a2p,...
        'activepositionproperty','position',...
        'XMinorTick','on',...
        'YMinorTick','on',...
        'FontSize',fontSize,...
        'color','w'...

```

```

);

set(hf1,'CurrentAxes',ha2);
Igdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'];
Igdlogic = true(length(Igdnames),1);
Igdhandles = [];
hl1=line( sdata(pm1).c1lineoutDist*1e3, ...
sdata(pm1).c1lineoutLowR, ...
'color','r','linewidth',1.0,'linestyle','--');
Igdhandles = [Igdhandles,hl1];
hl2=line( sdata(pm1).c1lineoutDist*1e3, ...
sdata(pm1).c1lineoutMidR, ...
'color','g','linewidth',1.0,'linestyle','--');
Igdhandles = [Igdhandles,hl2];
hl3=line( sdata(pm1).c1lineoutDist*1e3, ...
sdata(pm1).c1lineoutHighR, ...
'color','b','linewidth',1.0,'linestyle',':');
Igdhandles = [Igdhandles,hl3];
hl4=line( sdata(pm1).c1lineoutDist*1e3, ...
sdata(pm1).c1lineoutTotR, ...
'color','k','linewidth',1.0,'linestyle','--');
Igdhandles = [Igdhandles,hl4];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontSize',fontSize);
ylabel('\bfInverted Laser Transmission (AU)\rm','fontSize',fontSize);

title(['\bfParallel Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(Igdhandles,Igdnames(Igdlogic),'FontSize',fontSize);

IgdP = get(hlgd1,'Position');

ht1 = annotation('textbox',IgdP);

```

```

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 1',...
    'Image = 2',...
    'Side = Right',...
    ',...
    ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f) ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam1Cur,'%4.0f) ' kA')]...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

```

```

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-001_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'parProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'parProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 2 Left
if genParProC2L
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...

```

```

    'Visible','off...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hl1=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutLowL, ...
    'color','r','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutMidL, ...
    'color','g','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl2];
hl3=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutHighL, ...
    'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];
hl4=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutTotL, ...
    'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontSize',fontSize);
ylabel('\bfInverted Laser Transmission (AU)\rm','fontSize',fontSize);

title(['\bfParallel Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end

```

```

set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 2',...
    'Image = 2',...
    'Side = Left',...
    ',...
    ['Time = ' num2str(sdata(k).cam2Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam2Cur,'%4.0f') ' kA']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...

```

```

        -m3-a2ptight(2)-txtOff/figOutY...
    );
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-L'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'parProfiles');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'parProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 2 Right
if genParProC2R
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

```

```

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize',[figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hl1=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutLowR, ...
    'color','r','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutMidR, ...
    'color','g','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl2];
hl3=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutHighR, ...
    'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];
hl4=line( sdata(pm1).c2lineoutDist*1e3, ...
    sdata(pm1).c2lineoutTotR, ...
    'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[-inf,inf]);

xlabel('\bfDistance (mm)\rm','fontSize',fontSize);
ylabel('\bfInverted Laser Transmission (AU)\rm','fontSize',fontSize);

```

```
title(['\bfParallel Profile Lineout | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'], 'fontSize',fontSize);
```

```
grid on;
```

```
box on;
```

```
fytick = get(ha2,'ytick');
```

```
fytickstr = cell(1,length(fytick));
```

```
for j=1:length(fytick)
```

```
    fytickstr{j}=num2str(fytick(j), '%3.0f');
```

```
end
```

```
set(ha2,'yticklabel',fytickstr);
```

```
set(hf1,'CurrentAxes',ha1);
```

```
hlgd1=legend(lgdhandles,lgdnames(lgdlogic), 'FontSize',fontSize);
```

```
lgdp = get(hlgd1,'Position');
```

```
ht1 = annotation('textbox',lgdp);
```

```
set(ht1,'HorizontalAlignment','Left');
```

```
set(ht1,'VerticalAlignment','Top');
```

```
set(ht1,'Margin',0);
```

```
set(ht1,'LineStyle','none');
```

```
set(ht1,'FontSize',fontSize);
```

```
set(ht1,'String',...
```

```
    { ...
```

```
    '\bfShot Parameters\rm',...
```

```
    'Camera = 2',...
```

```
    'Image = 2',...
```

```
    'Side = Right',...
```

```
    ',...
```

```
    ['Time = ' num2str(sdata(k).cam2Time*1e9, '%4.0f') ' ns'],...
```

```
    ['Current = ' num2str(abs(sdata(k).cam2Cur), '%4.0f') ' kA']...
```

```
    });
```

```
set(ht1,'FitHeightToText','on');
```

```
set(ht1,'FitBoxToText','on');
```

```
tp = get(ht1,'position');
```

```
a2ptight = get(ha2,'tightinset');
```

```
a2pout = get(ha2,'outerposition');
```

```
m1 = tp(3) + 2*txtOff/figOutX;
```

```
m2 = lgdp(3) + 2*txtOff/figOutX;
```

```
m3 = txtOff/figOutY + a2ptight(4);
```

```

if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgd(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'parProfiles'];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'parProfiles']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

```

```

% Camera 1 Left k spectrum
if genParKProC1L
logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize',[figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off'...
    );

    ha2 = axes(...
        'Position',a2p,...
        'activepositionproperty','position',...
        'XMinorTick','on',...
        'YMinorTick','on',...
        'FontSize',fontSize,...
        'color','w'...
    );

    set(hf1,'CurrentAxes',ha2);
    lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'], 'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'};
    lgdlogic = true(length(lgdnames),1);
    lgdhandles = [];
    dmax = find((2*pi)/sdata(pm1).c1ksxLowL*1e3<=maxLambda,1,'first');
    hl1=line( ((2*pi)/sdata(pm1).c1ksxLowL)*1e3, ...
        (sdata(pm1).c1kMagLowL./max(sdata(pm1).c1kMagLowL(dmax:end))), ...
        'color','r','linewidth',1.0,'linestyle','--');
    lgdhandles = [lgdhandles,hl1];
    hl2=line( ((2*pi)/sdata(pm1).c1ksxMidL)*1e3, ...
        (sdata(pm1).c1kMagMidL./max(sdata(pm1).c1kMagMidL(dmax:end))), ...
        'color','g','linewidth',1.0,'linestyle','--');
    lgdhandles = [lgdhandles,hl2];
    hl3=line( ((2*pi)/sdata(pm1).c1ksxHighL)*1e3, ...
        (sdata(pm1).c1kMagHighL./max(sdata(pm1).c1kMagHighL(dmax:end))), ...

```

```

'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,h13];
h14=line( ((2*pi)./(sdata(pm1).c1ksxTotL)*1e3), ...
(sdata(pm1).c1kMagTotL./max(sdata(pm1).c1kMagTotL(dmax:end))), ...
'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,h14];

set(ha2,'xlim',[0,maxLambda]);
set(ha2,'ylim',[0,1]);

xlabel('\bfWavelength (mm)\rm','fontSize',fontSize);
ylabel('\bfNormalized Spectral Amplitude\rm','fontSize',fontSize);

title(['\bfWavelength Spectrum | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%4.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
{...
'\bfShot Parameters\rm',...
'Camera = 1',...
'Image = 2',...
'Side = Left',...
'',...
[Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f') ' ns'],...
[Current = ' num2str(abs(sdata(k).cam1Cur,'%4.0f') ' kA']...

```

```

    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-001_Image-002_Side-L'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\ ', 'fftSpectra']);

```

```

if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\ftSpectra']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 1 Right k spectrum
if genParKProC1R
logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off'...
    );

    ha2 = axes(...
        'Position',a2p,...
        'activepositionproperty','position',...
        'XMinorTick','on',...
        'YMinorTick','on',...
        'FontSize',fontSize,...
        'color','w'...
    );

    set(hf1,'CurrentAxes',ha2);

    lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'], 'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'};

```

```

lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
dmax = find((2*pi)./sdata(pm1).c1ksxLowR*1e3<=maxLambda,1,'first');
hl1=line( (2*pi)./(sdata(pm1).c1ksxLowR)*1e3, ...
    (sdata(pm1).c1kMagLowR./max(sdata(pm1).c1kMagLowR(dmax:end))), ...
    'color','r','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line( (2*pi)./(sdata(pm1).c1ksxMidR)*1e3, ...
    (sdata(pm1).c1kMagMidR./max(sdata(pm1).c1kMagMidR(dmax:end))), ...
    'color','g','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl2];
hl3=line( (2*pi)./(sdata(pm1).c1ksxHighR)*1e3, ...
    (sdata(pm1).c1kMagHighR./max(sdata(pm1).c1kMagHighR(dmax:end))), ...
    'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];
hl4=line( (2*pi)./(sdata(pm1).c1ksxTotR)*1e3, ...
    (sdata(pm1).c1kMagTotR./max(sdata(pm1).c1kMagTotR(dmax:end))), ...
    'color','k','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[0,maxLambda]);
set(ha2,'ylim',[0,1]);

xlabel('\bfWavelength (mm)\rm','fontSize',fontSize);
ylabel('\bfNormalized Spectral Amplitude\rm','fontSize',fontSize);

title(['\bfWavelength Spectrum | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%4.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');

```

```

set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 1',...
    'Image = 2',...
    'Side = Right',...
    },...
    ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam1Cur,'%4.0f') ' kA)]...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)];
lgdp = get(hlgd1,'position');

set(ht1,'position',[...

```

```

lgdp(1),...
lgdp(2)-txtOff/figOutY-tp(4),...
tp(3),...
tp(4));

filename = [ ...
'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
'_Camera-001_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\ ', 'fftSpectra'];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\fftSpectra']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 2 Left k spectrum
if genParKProc2L
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');
    set(gcf,'paperposition',[0,0,figOutX,figOutY]);
    set(gcf,'paperpositionmode','auto');
    set(gcf,'PaperSize', [figOutX,figOutY]);
    set(gcf,'units','inches');
    set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
    set(gcf,'units','normalized');

    ha1 = axes(...
        'Position',a1p,...
        'Visible','off...
    );

```

```

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);

lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'], 'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'];
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
dmax = find((2*pi)./sdata(pm1).c2ksxLowL*1e3<=maxLambda,1,'first');
hl1=line( (2*pi)./(sdata(pm1).c2ksxLowL)*1e3), ...
    (sdata(pm1).c2kMagLowL./max(sdata(pm1).c2kMagLowL(dmax:end))), ...
    'color','r','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line( (2*pi)./(sdata(pm1).c2ksxMidL)*1e3), ...
    (sdata(pm1).c2kMagMidL./max(sdata(pm1).c2kMagMidL(dmax:end))), ...
    'color','g','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl2];
hl3=line( (2*pi)./(sdata(pm1).c2ksxHighL)*1e3), ...
    (sdata(pm1).c2kMagHighL./max(sdata(pm1).c2kMagHighL(dmax:end))), ...
    'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];
hl4=line( (2*pi)./(sdata(pm1).c2ksxTotL)*1e3), ...
    (sdata(pm1).c2kMagTotL./max(sdata(pm1).c2kMagTotL(dmax:end))), ...
    'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[0,maxLambda]);
set(ha2,'ylim',[0,1]);

xlabel('\bfWavelength (mm)\rm','fontsize',fontSize);
ylabel('\bfNormalized Spectral Amplitude\rm','fontsize',fontSize);

title(['\bfWavelength Spectrum | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontsize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%4.1f');
end

```

```

set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm',...
    'Camera = 2',...
    'Image = 2',...
    'Side = Left',...
    ',...
    ['Time = ' num2str(sdata(k).cam2Time*1e9,'%4.0f') ' ns'],...
    ['Current = ' num2str(abs(sdata(k).cam2Cur,'%4.0f') ' kA']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...

```

```

        -m3-a2ptight(2)-txtOff/figOutY...
    );
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-L'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'fftSpectra'];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'fftSpectra']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

end

% Camera 2 Right k spectrum
if genParKProC2R
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

```

```

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize',[figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {[num2str(profileLow*100,'%3.0f') '% Threshold'],'Mean Position',[num2str(profileHigh*100,'%3.0f') '%
Threshold'],'Average'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
dmax = find((2*pi)/(sdata(pm1).c2ksxLowR*1e3)<=maxLambda,1,'first');
hl1=line( ((2*pi)/(sdata(pm1).c2ksxLowR)*1e3), ...
    (sdata(pm1).c2kMagLowR./max(sdata(pm1).c2kMagLowR(dmax:end))), ...
    'color','r','linewidth',1.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line( ((2*pi)/(sdata(pm1).c2ksxMidR)*1e3), ...
    (sdata(pm1).c2kMagMidR./max(sdata(pm1).c2kMagMidR(dmax:end))), ...
    'color','g','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl2];
hl3=line( ((2*pi)/(sdata(pm1).c2ksxHighR)*1e3), ...
    (sdata(pm1).c2kMagHighR./max(sdata(pm1).c2kMagHighR(dmax:end))), ...
    'color','b','linewidth',1.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];
hl4=line( ((2*pi)/(sdata(pm1).c2ksxTotR)*1e3), ...
    (sdata(pm1).c2kMagTotR./max(sdata(pm1).c2kMagTotR(dmax:end))), ...
    'color','k','linewidth',1.0,'linestyle','-');
lgdhandles = [lgdhandles,hl4];

set(ha2,'xlim',[0,maxLambda]);
set(ha2,'ylim',[0,1]);

xlabel('\bfWavelength (mm)\rm', 'fontSize',fontSize);

```

```

ylabel('\bfNormalized Spectral Amplitude\rm', 'fontSize', fontSize);

title(['\bfWavelength Spectrum | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'], 'fontSize', fontSize);

grid on;
box on;

fytick = get(ha2, 'ytick');
fytickstr = cell(1, length(fytick));
for j=1:length(fytick)
    fytickstr{j} = num2str(fytick(j), '%4.1f');
end
set(ha2, 'yticklabel', fytickstr);

set(hf1, 'CurrentAxes', ha1);

hlgd1 = legend(lgdhandles, lgdnames(lgdlogic), 'FontSize', fontSize);

lgdp = get(hlgd1, 'Position');

ht1 = annotation('textbox', lgdp);

set(ht1, 'HorizontalAlignment', 'Left');
set(ht1, 'VerticalAlignment', 'Top');
set(ht1, 'Margin', 0);
set(ht1, 'LineStyle', 'none');

set(ht1, 'FontSize', fontSize);
set(ht1, 'String', ...
    { ...
    '\bfShot Parameters\rm', ...
    'Camera = 2', ...
    'Image = 2', ...
    'Side = Right', ...
    }, ...
    ['Time = ' num2str(sdata(k).cam2Time*1e9, '%4.0f') ' ns'], ...
    ['Current = ' num2str(abs(sdata(k).cam2Cur), '%4.0f') ' kA']...
    ]);

set(ht1, 'FitHeightToText', 'on');
set(ht1, 'FitBoxToText', 'on');

tp = get(ht1, 'position');

a2ptight = get(ha2, 'tightinset');
a2pout = get(ha2, 'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;

```

```

m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Camera-002_Image-002_Side-R'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'fftSpectra');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\fftSpectra']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);
end

```

```

end

end

toc
end
% ===== end =====

% ===== start =====
% Plot Bulk Kinematics
if genBulkKinematics
display('Plotting Bulk Kinematics');

tsdata = [ ...
    [[sdata.cam1Time];[sdata.cam2Time]]*1e9, ...
    [[sdata.c1LowL];[sdata.c2LowL]]*1e3, ...
    [[sdata.c1MidL];[sdata.c2MidL]]*1e3, ...
    [[sdata.c1HighL];[sdata.c2HighL]]*1e3, ...
    [[sdata.c1LowR];[sdata.c2LowR]]*1e3, ...
    [[sdata.c1MidR];[sdata.c2MidR]]*1e3, ...
    [[sdata.c1HighR];[sdata.c2HighR]]*1e3, ... %7
    [[sdata.c1dLowL];[sdata.c2dLowL]]*1e3, ...
    [[sdata.c1dMidL];[sdata.c2dMidL]]*1e3, ...
    [[sdata.c1dHighL];[sdata.c2dHighL]]*1e3, ...
    [[sdata.c1dLowR];[sdata.c2dLowR]]*1e3, ...
    [[sdata.c1dMidR];[sdata.c2dMidR]]*1e3, ...
    [[sdata.c1dHighR];[sdata.c2dHighR]]*1e3, ... %13
    [[sdata.c1dLowUncL];[sdata.c2dLowUncL]]*1e3, ...
    [[sdata.c1dMidUncL];[sdata.c2dMidUncL]]*1e3, ...
    [[sdata.c1dHighUncL];[sdata.c2dHighUncL]]*1e3, ...
    [[sdata.c1dLowUncR];[sdata.c2dLowUncR]]*1e3, ...
    [[sdata.c1dMidUncR];[sdata.c2dMidUncR]]*1e3, ...
    [[sdata.c1dHighUncR];[sdata.c2dHighUncR]]*1e3, ... %19
    [[sdata.c1dLowVUncL];[sdata.c2dLowVUncL]]*1e-4, ...
    [[sdata.c1dMidVUncL];[sdata.c2dMidVUncL]]*1e-4, ...
    [[sdata.c1dHighVUncL];[sdata.c2dHighVUncL]]*1e-4, ...
    [[sdata.c1dLowVUncR];[sdata.c2dLowVUncR]]*1e-4, ...
    [[sdata.c1dMidVUncR];[sdata.c2dMidVUncR]]*1e-4, ...
    [[sdata.c1dHighVUncR];[sdata.c2dHighVUncR]]*1e-4, ... %25
    [[sdata.c1dLowAUncL];[sdata.c2dLowAUncL]], ...
    [[sdata.c1dMidAUncL];[sdata.c2dMidAUncL]], ...
    [[sdata.c1dHighAUncL];[sdata.c2dHighAUncL]], ...
    [[sdata.c1dLowAUncR];[sdata.c2dLowAUncR]], ...
    [[sdata.c1dMidAUncR];[sdata.c2dMidAUncR]], ...
    [[sdata.c1dHighAUncR];[sdata.c2dHighAUncR]], ... %31
    [[sdata.c1InitialTh];[sdata.c2InitialTh]]*1e3, ...
    [[sdata.c1InstantTh];[sdata.c2InstantTh]]*1e3, ...

```

```

[[sdata.c1InitialPos];[sdata.c2InitialPos]]*1e3, ...
[[sdata.c1InstantPos];[sdata.c2InstantPos]]*1e3, ... $35
[[sdata.c1InitialThUnc];[sdata.c2InitialThUnc]]*1e3, ...
[[sdata.c1InstantThUnc];[sdata.c2InstantThUnc]]*1e3, ...
[[sdata.c1InitialPosUnc];[sdata.c2InitialPosUnc]]*1e3, ...
[[sdata.c1InstantPosUnc];[sdata.c2InstantPosUnc]]*1e3 ... %39
];

logic1 = ~isnan(tsdata(:,2));

% left deltaX vs Time
if genPosLeft

hf1=figure('Visible','off');

a1p = [0,0,1,1,1];
a2p = [0,0,1,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

logic2 = tsdata(:,1)>=tAbl & logic1;
posTime = tsdata(logic2,1);
tvec = linspace(tAbl,max(tsdata(logic2,1)),100);

pos = tsdata(logic2,8);
Unc = tsdata(logic2,14);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...

```

```

    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) ')^2'],'options',fopt);
[fdata1,gof1,fout1]=fit(posTime,pos,ftype);
fconf1=confint(fdata1,fitConf);
fitData1 = fdata1.a*(tvec-tAbl)+0.5*fdata1.b*(tvec-tAbl).^2;

pos = tsdata(logic2,9);
Unc = tsdata(logic2,15);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) ')^2'],'options',fopt);
[fdata2,gof2,fout2]=fit(posTime,pos,ftype);
fconf2=confint(fdata2,fitConf);
fitData2 = fdata2.a*(tvec-tAbl)+0.5*fdata2.b*(tvec-tAbl).^2;

pos = tsdata(logic2,10);
Unc = tsdata(logic2,16);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) ')^2'],'options',fopt);
[fdata3,gof3,fout3]=fit(posTime,pos,ftype);
fconf3=confint(fdata3,fitConf);
fitData3 = fdata3.a*(tvec-tAbl)+0.5*fdata3.b*(tvec-tAbl).^2;

set(hf1,'CurrentAxes',ha2);
lgdnames = {[num2str(profileLow*100,%3.0f) '% Position'],'Mean Position',[num2str(profileHigh*100,%3.0f) '% Position'],...
    [num2str(profileLow*100,%3.0f) '% Position Accel.'],'Mean Position Accel.',[num2str(profileHigh*100,%3.0f) '% Position
    Accel.']}];
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,8),tsdata(logic1,14),'r','linestyle','none', ...
    'marker','s','markeredgecolor','k','markerfacecolor','r');
lgdhandles = [lgdhandles,hl1];
hl2=errorbar( tsdata(logic1,1),tsdata(logic1,9),tsdata(logic1,15),'g','linestyle','none', ...
    'marker','o','markeredgecolor','k','markerfacecolor','g');
lgdhandles = [lgdhandles,hl2];
hl3=errorbar( tsdata(logic1,1),tsdata(logic1,10),tsdata(logic1,16),'b','linestyle','none', ...
    'marker','d','markeredgecolor','k','markerfacecolor','b');
lgdhandles = [lgdhandles,hl3];
hl5=line( tvec,fitData1,'linestyle','--','color','r','linewidth',2);
lgdhandles = [lgdhandles,hl5];
hl6=line( tvec,fitData2,'linestyle','--','color','g','linewidth',2);
lgdhandles = [lgdhandles,hl6];
hl7=line( tvec,fitData3,'linestyle','--','color','b','linewidth',2);

```

```

    lgdhandles = [lgdhandles,h17];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfDistance Traveled (mm)\rm','fontSize',fontSize);

title(['\bfLeft Side Expansion | ' series ' Offset | +/- ' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'fontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
%{
set(ht1,'String',...
    { ...
    [num2str(profileLow*100,'%3.0f') '% Position Acceleration'],...
    'Model: x = v_{fixed} * t + 1/2 a * t^2',...
    ['v_{fixed} = ' num2str(1e2*-vAbl,'%5.2f') ' (cm/\mus)],...
    ['a = ' num2str(1e5*fdata1.a,'%5.2f') '+/-' num2str(1e5*(fdata1.a-fconf1(1,1)),'%5.2f') ' (cm/\mus^2)],...
    ['Confidence = ' num2str(fitConf*100,'%2.0f') '%'],...
    ['R Squared = ' num2str(gof1.rsquare,'%4.2f)],...
    "....
    'Mean Position Acceleration',...
    'Model: x = v_{fixed} * t + 1/2 a * t^2',...
    ['v_{fixed} = ' num2str(1e2*-vAbl,'%5.2f') ' (cm/\mus)],...
    ['a = ' num2str(1e5*fdata2.a,'%5.2f') '+/-' num2str(1e5*(fdata2.a-fconf2(1,1)),'%5.2f') ' (cm/\mus^2)],...
    ['Confidence = ' num2str(fitConf*100,'%2.0f') '%'],...

```

```

['R Squared = ' num2str(gof2.rsquare,%4.2f)],...
"...
[num2str(profileHigh*100,%3.0f) '% Position Acceleration'],...
'Model: x = v_{fixed} * t + 1/2 a * t^2',...
['v_{fixed} = ' num2str(1e2*-vAbl,%5.2f) ' (cm/\mus)],...
['a = ' num2str(1e5*fdata3.a,%5.2f) '+/-' num2str(1e5*(fdata3.a-fconf3(1,1)),%5.2f) ' (cm/\mus^2)],...
['Confidence = ' num2str(fitConf*100,%2.0f) '% ],...
['R Squared = ' num2str(gof3.rsquare,%4.2f)],...
"...
});
%}
set(ht1,'String',...
{...
[num2str(profileLow*100,%3.0f) '% Position Acceleration'],...
'Model: x = v * t + 1/2 a * t^2',...
['v = ' num2str(1e2*fdata1.a,%5.2f) '+/-' num2str(1e2*(fdata1.a-fconf1(1,1)),%5.2f) ' (cm/\mus)],...
['a = ' num2str(1e5*fdata1.b,%5.2f) '+/-' num2str(1e5*(fdata1.b-fconf1(1,2)),%5.2f) ' (cm/\mus^2)],...
['Confidence = ' num2str(fitConf*100,%2.0f) '% ],...
['R Squared = ' num2str(gof1.rsquare,%4.2f)],...
"...
'Mean Position Acceleration',...
'Model: x = v * t + 1/2 a * t^2',...
['v = ' num2str(1e2*fdata2.a,%5.2f) '+/-' num2str(1e2*(fdata2.a-fconf2(1,1)),%5.2f) ' (cm/\mus)],...
['a = ' num2str(1e5*fdata2.b,%5.2f) '+/-' num2str(1e5*(fdata2.b-fconf2(1,2)),%5.2f) ' (cm/\mus^2)],...
['Confidence = ' num2str(fitConf*100,%2.0f) '% ],...
['R Squared = ' num2str(gof2.rsquare,%4.2f)],...
"...
[num2str(profileHigh*100,%3.0f) '% Position Acceleration'],...
'Model: x = v * t + 1/2 a * t^2',...
['v = ' num2str(1e2*fdata3.a,%5.2f) '+/-' num2str(1e2*(fdata3.a-fconf3(1,1)),%5.2f) ' (cm/\mus)],...
['a = ' num2str(1e5*fdata3.b,%5.2f) '+/-' num2str(1e5*(fdata3.b-fconf3(1,2)),%5.2f) ' (cm/\mus^2)],...
['Confidence = ' num2str(fitConf*100,%2.0f) '% ],...
['R Squared = ' num2str(gof3.rsquare,%4.2f)],...
"...
});

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...

```

```

        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgd(4),...
    lgd(3),lgd(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgd(1),...
    lgd(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ distFile '_Series-' series '_Side-Left' ];

print(hf1, '-diff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'position');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\position']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% right deltaX vs Time
if genPosRight

hf1=figure('Visible','off');
```

```

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize',[figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

logic2 = tsdata(:,1)>=tAbl & logic1;
posTime = tsdata(logic2,1);
tvec = linspace(tAbl,max(tsdata(logic2,1)),100);

pos = tsdata(logic2,11);
Unc = tsdata(logic2,17);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fittype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) ')^2'],'options',fopt);
[fdata1,gof1,fout1]=fit(posTime,pos,ftype);
fconf1=confint(fdata1,fitConf);
fitData1 = fdata1.a*(tvec-tAbl)+0.5*fdata1.b*(tvec-tAbl).^2;

pos = tsdata(logic2,12);
Unc = tsdata(logic2,18);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fittype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) ')^2'],'options',fopt);
[fdata2,gof2,fout2]=fit(posTime,pos,ftype);

```

```

fconf2=confint(fdata2,fitConf);
fitData2 = fdata2.a*(tvec-tAbl)+0.5*fdata2.b*(tvec-tAbl).^2;

pos = tsdata(logic2,13);
Unc = tsdata(logic2,19);
wgt = Unc.^(-2);
fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',posFitLow,'Upper',posFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fittype(['a*(x-' num2str(tAbl) ')'+0.5*b*(x-' num2str(tAbl) )^2'],'options',fopt);
[fdata3,gof3,fout3]=fit(posTime,pos,ftype);
fconf3=confint(fdata3,fitConf);
fitData3 = fdata3.a*(tvec-tAbl)+0.5*fdata3.b*(tvec-tAbl).^2;

set(hf1,'CurrentAxes',ha2);
Igdnames = {[num2str(profileLow*100,%3.0f) '% Position'],'Mean Position',[num2str(profileHigh*100,%3.0f) '% Position'],...
    [num2str(profileLow*100,%3.0f) '% Position Accel.'],'Mean Position Accel.',[num2str(profileHigh*100,%3.0f) '% Position Accel.']}];
Igdlogic = true(length(Igdnames),1);
Igdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,11),tsdata(logic1,17),'r','linestyle','none', ...
    'marker','s','markeredgecolor','k','markerfacecolor','r');
Igdhandles = [Igdhandles,hl1];
hl2=errorbar( tsdata(logic1,1),tsdata(logic1,12),tsdata(logic1,18),'g','linestyle','none', ...
    'marker','o','markeredgecolor','k','markerfacecolor','g');
Igdhandles = [Igdhandles,hl2];
hl3=errorbar( tsdata(logic1,1),tsdata(logic1,13),tsdata(logic1,19),'b','linestyle','none', ...
    'marker','d','markeredgecolor','k','markerfacecolor','b');
Igdhandles = [Igdhandles,hl3];
hl5=line( tvec,fitData1,'linestyle','--','color','r','linewidth',2);
Igdhandles = [Igdhandles,hl5];
hl6=line( tvec,fitData2,'linestyle','--','color','g','linewidth',2);
Igdhandles = [Igdhandles,hl6];
hl7=line( tvec,fitData3,'linestyle','--','color','b','linewidth',2);
Igdhandles = [Igdhandles,hl7];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfDistance Traveled (mm)\rm','fontSize',fontSize);

title(['\bfRight Side Expansion | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');

```

```

fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    [num2str(profileLow*100,'%3.0f') '% Position Acceleration'],...
    'Model:  $x = v * t + 1/2 a * t^2$ ',...
    ['v = ' num2str(1e2*fdata1.a,'%5.2f') '+/-' num2str(1e2*(fdata1.a-fconf1(1,1)),'%5.2f') ' (cm/\mus)'],...
    ['a = ' num2str(1e5*fdata1.b,'%5.2f') '+/-' num2str(1e5*(fdata1.b-fconf1(1,2)),'%5.2f') ' (cm/\mus^2)'],...
    ['Confidence = ' num2str(fitConf*100,'%2.0f') '%'],...
    ['R Squared = ' num2str(gof1.rsquare,'%4.2f)'],...
    "...
    'Mean Position Acceleration',...
    'Model:  $x = v * t + 1/2 a * t^2$ ',...
    ['v = ' num2str(1e2*fdata2.a,'%5.2f') '+/-' num2str(1e2*(fdata2.a-fconf2(1,1)),'%5.2f') ' (cm/\mus)'],...
    ['a = ' num2str(1e5*fdata2.b,'%5.2f') '+/-' num2str(1e5*(fdata2.b-fconf2(1,2)),'%5.2f') ' (cm/\mus^2)'],...
    ['Confidence = ' num2str(fitConf*100,'%2.0f') '%'],...
    ['R Squared = ' num2str(gof2.rsquare,'%4.2f)'],...
    "...
    [num2str(profileHigh*100,'%3.0f') '% Position Acceleration'],...
    'Model:  $x = v * t + 1/2 a * t^2$ ',...
    ['v = ' num2str(1e2*fdata3.a,'%5.2f') '+/-' num2str(1e2*(fdata3.a-fconf3(1,1)),'%5.2f') ' (cm/\mus)'],...
    ['a = ' num2str(1e5*fdata3.b,'%5.2f') '+/-' num2str(1e5*(fdata3.b-fconf3(1,2)),'%5.2f') ' (cm/\mus^2)'],...
    ['Confidence = ' num2str(fitConf*100,'%2.0f') '%'],...
    ['R Squared = ' num2str(gof3.rsquare,'%4.2f)'],...
    "...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

```

```

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ distFile '_Series-' series '_Side-Right' ];

print(hf1, '-d.tif', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'position'];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\position']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end
end

```

```

close(hf1);

end

% left <velocity> vs Time
if genVelLeft

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

posTime = tsdata(logic1,1);
vel = tsdata(logic1,9)./tsdata(logic1,1);
Unc = tsdata(logic1,21);
wgt = Unc.^(-2);

fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',velFitLow,'Upper',velFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype('a+b*x','options',fopt);
[fdata,gof,fout]=fit(posTime,vel,ftype);

fconf=confint(fdata,fitConf);

tvec = linspace(min(tsdata(logic1,1)),max(tsdata(logic1,1)),100);

```

```

linFitData = fdata.a+fdata.b*tvec;

set(hf1,'CurrentAxes',ha2);

lgdnames = {[num2str(profileLow*100,'%3.0f') % Position'],'Mean Position',[num2str(profileHigh*100,'%3.0f') %
Position'],'Regression'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,8)./tsdata(logic1,1)*100,tsdata(logic1,20),'r','linestyle','none', ...
'marker','s','markeredgecolor','k','markerfacecolor','r');
lgdhandles = [lgdhandles,hl1];
hl2=errorbar( tsdata(logic1,1),tsdata(logic1,9)./tsdata(logic1,1)*100,tsdata(logic1,21),'g','linestyle','none', ...
'marker','o','markeredgecolor','k','markerfacecolor','g');
lgdhandles = [lgdhandles,hl2];
hl3=errorbar( tsdata(logic1,1),tsdata(logic1,10)./tsdata(logic1,1)*100,tsdata(logic1,22),'b','linestyle','none', ...
'marker','d','markeredgecolor','k','markerfacecolor','b');
lgdhandles = [lgdhandles,hl3];
hl4=line( tvec,linFitData*100,'linestyle','--','color','g');
lgdhandles = [lgdhandles,hl4];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontsize',fontSize);
ylabel('\bf<Velocity> (cm/\mus)\rm','fontsize',fontSize);

title(['\bfLeft Side Expansion Velocity | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'],'fontsize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);

```

```

set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    'Model: v = v_{0} + a * t',...
    ['v_{0} = ' num2str(1e2*fdata.a,%5.2f) '+/-' num2str(1e2*(fdata.a-fconf(1,1)),%5.2f) ' (cm/\mus)],...
    ['a = ' num2str(1e5*fdata.b,%5.2f) '+/-' num2str(1e5*(fdata.b-fconf(1,2)),%5.2f) ' (cm/\mus^2)],...
    ['Confidence = ' num2str(fitConf*100,%2.0f) '%'],...
    ['R Squared = ' num2str(gof.rsquare,%4.2f)]...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...

```

```

tp(4)];

filename = [ velFile '_Series-' series '_Side-Left' ];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'velAve');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\velAve']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% right <velocity> vs Time
if genVelRight

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
    );

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...

```

```

);

posTime = tsdata(logic1,1);
vel = tsdata(logic1,12)/tsdata(logic1,1);
Unc = tsdata(logic1,24);
wgt = Unc.^(-2);

fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',velFitLow,'Upper',velFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype('a+b*x','options',fopt);
[fdata,gof,fout]=fit(posTime,vel,ftype);

fconf=confint(fdata,fitConf);

tvec = linspace(min(tsdata(logic1,1)),max(tsdata(logic1,1)),100);
linFitData = fdata.a+fdata.b*tvec;

set(hf1,'CurrentAxes',ha2);
lgdnames = {[num2str(profileLow*100,'%3.0f') % Position'],'Mean Position',[num2str(profileHigh*100,'%3.0f') %
Position'],'Regression'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,11)/tsdata(logic1,1)*100,tsdata(logic1,23),'r','linestyle','none', ...
    'marker','s','markeredgecolor','k','markerfacecolor','r');
    lgdhandles = [lgdhandles,hl1];
hl2=errorbar( tsdata(logic1,1),tsdata(logic1,12)/tsdata(logic1,1)*100,tsdata(logic1,24),'g','linestyle','none', ...
    'marker','o','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,hl2];
hl3=errorbar( tsdata(logic1,1),tsdata(logic1,13)/tsdata(logic1,1)*100,tsdata(logic1,25),'b','linestyle','none', ...
    'marker','d','markeredgecolor','k','markerfacecolor','b');
    lgdhandles = [lgdhandles,hl3];
hl4=line( tvec,linFitData*100,'linestyle','--','color','g');
    lgdhandles = [lgdhandles,hl4];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bf<Velocity> (cm/\mus)\rm','fontSize',fontSize);

title(['\bfRight Side Expansion Velocity | ' series ' Offset | +/- ' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));

```

```

for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j), '%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    'Model: v = v_{0} + a * t',...
    ['v_{0} = ' num2str(1e2*fdata.a, '%5.2f') ' +/- ' num2str(1e2*(fdata.a-fconf(1,1)), '%5.2f') ' (cm/\mus)',...
    ['a = ' num2str(1e5*fdata.b, '%5.2f') ' +/- ' num2str(1e5*(fdata.b-fconf(1,2)), '%5.2f') ' (cm/\mus^2)',...
    ['Confidence = ' num2str(fitConf*100, '%2.0f') '%'],...
    ['R Squared = ' num2str(gof.rsquare, '%4.2f')']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...

```

```

        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    );
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ velFile '_Series-' series '_Side-Right' ];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'velAve');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\velAve']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% left amplitude vs Time
if genAmpLeft

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);

```

```

set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

ampTime = tsdata(logic1,1);
amp = abs(tsdata(logic1,10)-tsdata(logic1,8));
Unc = sqrt(tsdata(logic1,16).^2+tsdata(logic1,14).^2);
wgt = Unc.^(-2);
tvec = linspace(0,200,100);

fopt=fitoptions('method',fitMethod,'Lower',ampFitLow,'Upper',ampFitHigh,'startpoint',[.1,1e-3],'weights',wgt);
ftype=fitype('a*exp(b*x)','options',fopt);
[fdata,gof]=fit(ampTime,amp,ftype);

fconf=confint(fdata,fitConf);
expFitData = fdata.a*exp(fdata.b*tvec);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Amplitude','Exponential Fit'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( ampTime, amp, Unc,'k', ...
    'linestyle','none','marker','s','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,hl1];
hl2=line( tvec, expFitData, ...
    'linestyle','--','color','c','linewidth',2.0);
    lgdhandles = [lgdhandles,hl2];
hold off

set(ha2,'xlim',[0,200]);
set(ha2,'ylim',[0,ampUp]);

xlabel('\bfTime (ns)rm','fontSize',fontSize);
ylabel('\bfAmplitude F.O.M. (mm)rm','fontSize',fontSize);

```

```

title(['\bfLeft Side Instability Growth | ' series ' Offset | +/- ' num2str(params.vCharge(logicP)) ' kV\rm'], 'fontsize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%4.2F');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    'Model: FOM = A * exp( \gamma t )',...
    ['A = ' num2str(fdata.a, '%5.2f') ' +/- ' num2str(fdata.a-fconf(1,1), '%5.2f') ' (mm)'],...
    ['\gamma = ' num2str(fdata.b*1e3, '%4.0f') ' +/- ' num2str((fdata.b-fconf(1,2))*1e3, '%4.0f') ' (\mus^{-1})'],...
    ['Confidence = ' num2str(fitConf*100, '%2.0f') '%'],...
    ['R Squared = ' num2str(gof.rsquare, '%4.2f')'],...
    ",...
    'Measured e-Folding Time',...
    ['1/\gamma = ' num2str(1/(fdata.b), '%4.0f') ' +/- ' num2str(sqrt((fdata.b-fconf(1,2))^2/(fdata.b)^4), '%4.0f') ' (ns)']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;

```

```

m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)];
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ampFile '_Series-' series '_Side-Left' ];

print(hf1, '-d TIFF', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'amplitude');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\amplitude']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% right amplitude vs Time

```

```

if genAmpRight

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

ampTime = tsdata(logic1,1);
amp = abs(tsdata(logic1,13)-tsdata(logic1,11));
Unc = sqrt(tsdata(logic1,19).^2+tsdata(logic1,17).^2);
wgt = Unc.^(-2);

fopt=fitoptions('method',fitMethod,'Lower',ampFitLow,'Upper',ampFitHigh,'startpoint',[.1,1e-3],'weights',wgt);
ftype=fitype('a*exp(b*x)','options',fopt);
[fdata,gof]=fit(ampTime,amp,ftype);

fconf=confint(fdata,fitConf);

%tvec = linspace(min(tsdata(logic1,1)),max(tsdata(logic1,1)),100);
tvec = linspace(0,200,100);
expFitData = fdata.a*exp(fdata.b*tvec);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Amplitude','Exponential Fit'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on

```

```

hl1=errorbar( tdata(logic1,1), amp, Unc,'k', ...
    'linestyle','none','marker','s','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,hl1];
hl2=line( tvec, expFitData, ...
    'linestyle','--','color','c','linewidth',2.0);
    lgdhandles = [lgdhandles,hl2];
hold off

set(ha2,'xlim',[0,200]);
set(ha2,'ylim',[0,ampUp]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfAmplitude F.O.M. (mm)\rm','fontSize',fontSize);

title(['\bfRight Side Instability Growth | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),%4.2f);
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    'Model: FOM = A * exp( \gamma t )',...
    ['A = ' num2str(fdata.a,%5.2f) '+/-' num2str(fdata.a-fconf(1,1),%5.2f) ' (mm)'],...
    ['\gamma = ' num2str(fdata.b*1e3,%4.0f) '+/-' num2str((fdata.b-fconf(1,2))*1e3,%4.0f) ' (\mus^{-1})'],...
    ['Confidence = ' num2str(fitConf*100,%2.0f) '%'],...
    ['R Squared = ' num2str(gof.rsquare,%4.2f)],...
    "...
    'Measured e-Folding Time',...

```

```

[1/\gamma = ' num2str(1/(fdata.b),%4.0f) '+/-' num2str(sqrt((fdata.b-fconf(1,2))^2/(fdata.b)^4),%4.0f) ' (ns)']...
});

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ampFile '_Series-' series '_Side-Right' ];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'amplitude');
if s==1

```

```

    movefile( [filename '.tif'], ...
             [sumPath '\amplitude']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% initial thickness
if genInitialTh

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
    );

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
    );

inTh = tsdata(logic1,32);
Unc = tsdata(logic1,36);
wgt = Unc.^(-2);

inThMean = sum(wgt.*inTh)/sum(wgt);
inThUncM = 1/sqrt(sum(wgt));
inThStd = sqrt(1/(length(inTh)-1)*sum(wgt.*(inTh-(sum(inTh.*wgt)/sum(wgt))).^2)*length(inTh)/sum(wgt));

```

```

[inThCount,inThBinVals] = hist(inTh,linspace(inThBinLow,inThBinHigh,inThBinNum));

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Distribution'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hb1=bar(inThBinVals,inThCount,'edgecolor','k','facecolor','g','linewidth',2);
    lgdhandles = [lgdhandles,hb1];

hold off

set(ha2,'xlim',[0,inf]);
set(ha2,'ylim',[0,max(inThCount)+1]);

xlabel('\bfObserved Thickness (mm)\rm','fontSize',fontSize);
ylabel('\bfObservations\rm','fontSize',fontSize);

title(['\bfInitial Foil Thickness | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    'Mean',...
    ['W = ' num2str(inThMean,'%4.2f') '+/-' num2str(inThUncM,'%4.2f') ' (mm)],...

```

```

"...
'Standard Deviation',...
['\sigma = +/-' num2str(inThStd,'%4.2f) ' (mm)']...
});

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ thickFile '_Series-' series '_Initial' ];

print(hf1, '-diff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...

```

```

    [sumPath '\], 'thickness');
if s==1
    movefile( [filename '.tif'], ...
    [sumPath '\thickness']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% initial position
if genInitialPos

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
    );

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
    );

inPos = tsdata(logic1,34);
Unc = tsdata(logic1,38);
wgt = Unc.^(-2);

inPosMean = sum(wgt.*inPos)/sum(wgt);

```

```

inPosUncM = 1/sqrt(sum(wgt));
inPosStd = sqrt(1/(length(inPos)-1)*sum(wgt.*(inPos-(sum(inPos.*wgt)/sum(wgt))).^2)*length(inPos)/sum(wgt));

[inPosCount,inPosBinVals] = hist(inPos,linspace(inPosBinLow,inPosBinHigh,inPosBinNum));

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Distribution'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hb1=bar(inPosBinVals,inPosCount,'edgecolor','k','facecolor','g','linewidth',2);
    lgdhandles = [lgdhandles,hb1];

hold off

set(ha2,'xlim',[-inf,inf]);
set(ha2,'ylim',[0,max(inPosCount)+1]);

xlabel('\bfObserved Position (mm)\rm','fontSize',fontSize);
ylabel('\bfObservations\rm','fontSize',fontSize);

title(['\bfInitial Foil Position | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...

```

```

'Mean',...
['X_{0} = ' num2str(inPosMean,'%4.2f') '+/-' num2str(inPosUncM,'%4.2f') ' (mm)'],...
"...
'Standard Deviation',...
['\sigma = +/-' num2str(inPosStd,'%4.2f') ' (mm)']...
});

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ distFile '_Series-' series '_Initial' ];

print(hf1, '-diff', ['-r' num2str(pltres)], filename);

```

```

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'position');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\position']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% instant thickness vs Time
if genInstantTh

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
    );

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
    );

posTime = tsdata(logic1,1);
pos = tsdata(logic1,33);
Unc = tsdata(logic1,37);

```

```

wgt = Unc.^(-2);

fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',thCenFitLow,'Upper',thCenFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
ftype=fitype('a+b*x','options',fopt);
[fdata,gof,fout]=fit(posTime,pos,ftype);

fconf=confint(fdata,fitConf);

tvec = linspace(min(tsdata(logic1,1)),max(tsdata(logic1,1)),100);
linFitData = fdata.a+fdata.b*tvec;

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Thickness'}; %,'Linear Fit'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,33),tsdata(logic1,37),'k','linestyle','none', ...
    'marker','o','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,hl1];
%hl2=line( tvec,linFitData,'linestyle','--','color','g','linewidth',2.0);
%    lgdhandles = [lgdhandles,hl2];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfObserved Thickness (mm)\rm','fontSize',fontSize);

title(['\bfBulk Plasma Thickness | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'],'fontSize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

```

```

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String','');
% {
set(ht1,'String',...
    { ...
    'Model: \Delta = \Delta_{0} + v * t',...
    ['\Delta_{0} = ' num2str(fdata.a,%5.2f) '+/-' num2str(fdata.a-fconf(1,1),%5.2f) ' (mm)],...
    ['v = ' num2str(100*fdata.b,%5.2f) '+/-' num2str(100*(fdata.b-fconf(1,2)),%5.2f) ' (cm/\mus)],...
    ['Confidence = ' num2str(fitConf*100,%2.0f) '%'],...
    ['R Squared = ' num2str(gof.rsquare,%4.2f)]...
    });
% }
set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);

```

```

lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ thickFile '_Series-' series '_Instant' ];

print(hf1, '-drtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'thickness');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\thickness']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% instant position vs Time
if genInstantPos

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off...
    );

ha2 = axes(...

```

```

'Position',a2p,...
'activepositionproperty','position',...
'XMinorTick','on',...
'YMinorTick','on',...
'FontSize',fontSize,...
'color','w'...
);

posTime = tsdata(logic1,1);
pos = tsdata(logic1,35);
Unc = tsdata(logic1,39);
wgt = Unc.^(-2);

fopt=fitoptions('method','nonlinearleastsquares',...
    'Lower',thCenFitLow,'Upper',thCenFitHigh,...
    'startpoint',[0,0],'weights',wgt,'MaxFunEvals',fitIterate,'MaxIter',fitIterate);
fitype=fitype('a+b*x','options',fopt);
[fdata,gof,fout]=fit(posTime,pos,fitype);

fconf=confint(fdata,fitConf);

tvec = linspace(min(tsdata(logic1,1)),max(tsdata(logic1,1)),100);
linFitData = fdata.a+fdata.b*tvec;

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Position'}; % , Linear Fit';
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on
hl1=errorbar( tsdata(logic1,1),tsdata(logic1,35),tsdata(logic1,39),'k','linestyle','none', ...
    'marker','o','markeredgecolor','k','markerfacecolor','g');
    lgdhandles = [lgdhandles,hl1];
%hl2=line( tvec,linFitData,'linestyle','--','color','g','linewidth',2.0);
%    lgdhandles = [lgdhandles,hl2];
hold off

set(ha2,'xlim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontsize',fontSize);
ylabel('\bfObserved Position (mm)\rm','fontsize',fontSize);

title(['\bfBulk Plasma Position | ' series ' Offset | +/-' num2str(params.vCharge(logicP)) ' kV\rm'], 'fontsize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)

```

```

    fytickstr{j}=num2str(fytick(j),'%3.1f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),FontSize,fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String','');
% {
set(ht1,'String',...
    { ...
    'Model:  $x = x_{0} + v * t$ ,...
    ['x_{0} = ' num2str(fdata.a,%5.2f) '+/-' num2str(fdata.a-fconf(1,1),%5.2f) ' (mm)],...
    ['v = ' num2str(100*fdata.b,%5.2f) '+/-' num2str(100*(fdata.b-fconf(1,2)),%5.2f) ' (cm/mus)],...
    ['Confidence = ' num2str(fitConf*100,%2.0f) '%'],...
    ['R Squared = ' num2str(gof.rsquare,%4.2f)]...
    });
% }
set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...

```

```

        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ distFile '_Series-' series '_Instant' ];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'position');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\position']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

toc
end
% ===== end =====

% ===== start =====
% Plot Annotated Images
if genAnnImages

display('Generating Annotated Images');

```

```

for k=1:snun

    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    % Camera 1
    logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
    if sum(logicP1) > 0
        logic2 = imap(:,3)==params.cam1ID(logicP1);
        logic3 = imap(:,4)==params.cam1PreID(logicP1);
        logic4 = imap(:,4)==params.cam1ImID(logicP1);
        timap = imap(logic1 & logic2 & logic4,:);

        [image,temp] = load_images(timap(1,2:4),imTType,shotPath);

        hf1=figure('Visible','off');

        a1p = [0,0,1,1];
        a2p = [0,0,1,1];

        set(gcf,'paperunits','inches');
        set(gcf,'paperposition',[0,0,figOutX,figOutY]);
        set(gcf,'paperpositionmode','auto');
        set(gcf,'PaperSize',[figOutX,figOutY]);
        set(gcf,'units','inches');
        set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
        set(gcf,'units','normalized');

        ha1 = axes(...
            'Position',a1p,...
            'Visible','off...
        );

        ha2 = axes(...
            'Position',a2p,...
            'activepositionproperty','position',...
            'XMinorTick','on',...
            'YMinorTick','on',...
            'FontSize',fontSize,...
            'color','w'...
        );

        set(hf1,'CurrentAxes',ha2);

        lgdnames = {'Anode','Midplane','Foil Initial Position','ROI', ...
            [num2str(profileLow*100,'%3.0f') '% Position'],'Mean Position', ...
            [num2str(profileHigh*100,'%3.0f') '% Position']};

```

```

if sideText
    lgdlogic = true(length(lgdnames),1);
    infoText = { ...
        '\bfShot Parameters\rm',...
        ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f) ' ns'],...
        ['Current = ' num2str(abs(sdata(k).cam1Cur),'%4.0f) ' kA']...
    };
else
    lgdlogic = false(length(lgdnames),1);
    infoText = "";
end

%lgdlogic = true(length(lgdnames),1);
lgdhandles = [];

imagesc(image.data(:,,:2));
colormap(gray);
hold on;

axis([sdata(pm1).c1pxl-25,sdata(pm1).c1pxr+25,-inf,inf])

% anode boundaries
hl1 = line( ...
    [sdata(pm1).c1pxl,sdata(pm1).c1pxl], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','c','linewidth',2.0);
hl2 = line( ...
    [sdata(pm1).c1pxr,sdata(pm1).c1pxr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','c','linewidth',2.0);
lgdhandles = [lgdhandles,hl2];

% midplane
hl3 = line( ...
    [sdata(pm1).c1MidDistInd,sdata(pm1).c1MidDistInd], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','c','linewidth',2.0,'linestyle','--');
lgdhandles = [lgdhandles,hl3];

% initial foil boundaries
hl4 = line( ...
    [sdata(pm1).c1pfxl,sdata(pm1).c1pfxl], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','y','linewidth',1.0);
hl5 = line( ...
    [sdata(pm1).c1pfxr,sdata(pm1).c1pfxr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','y','linewidth',1.0);
lgdhandles = [lgdhandles,hl5];

```

```

% left roi boundaries
hl6 = line( ...
    [sdata(pm1).c1lxl,sdata(pm1).c1lxl], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','m','linewidth',1.0);
hl7 = line( ...
    [sdata(pm1).c1lxl,sdata(pm1).c1lxl], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','m','linewidth',1.0);
hl8 = line( ...
    [sdata(pm1).c1lxl,sdata(pm1).c1lxr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yt], ...
    'Color','m','linewidth',1.0);
hl9 = line( ...
    [sdata(pm1).c1lxl,sdata(pm1).c1lxr], ...
    [sdata(pm1).c1yb,sdata(pm1).c1yb], ...
    'Color','m','linewidth',1.0);
lgdhandles = [lgdhandles,hl9];

% left low threshold boundaries
hl10 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileLowIndL+lineoutAvr,sdata(pm1).c1profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','r','linewidth',0.5);
hl11 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileLowIndL-lineoutAvr,sdata(pm1).c1profileLowIndL-lineoutAvr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','r','linewidth',0.5);
hl12 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileLowIndL-lineoutAvr,sdata(pm1).c1profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yt], ...
    'Color','r','linewidth',0.5);
hl13 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileLowIndL-lineoutAvr,sdata(pm1).c1profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c1yb,sdata(pm1).c1yb], ...
    'Color','r','linewidth',0.5);
lgdhandles = [lgdhandles,hl13];

% left mid threshold boundaries
hl14 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileMidIndL+lineoutAvr,sdata(pm1).c1profileMidIndL+lineoutAvr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','g','linewidth',0.5);
hl15 = line( ...
    sdata(pm1).c1lxl+[sdata(pm1).c1profileMidIndL-lineoutAvr,sdata(pm1).c1profileMidIndL-lineoutAvr], ...
    [sdata(pm1).c1yt,sdata(pm1).c1yb], ...
    'Color','g','linewidth',0.5);
hl16 = line( ...

```

```

sdata(pm1).c1lxl+[sdata(pm1).c1profileMidIndL-lineoutAvr,sdata(pm1).c1profileMidIndL+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','g','linewidth',0.5);
hl17 = line( ...
sdata(pm1).c1lxl+[sdata(pm1).c1profileMidIndL-lineoutAvr,sdata(pm1).c1profileMidIndL+lineoutAvr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...
'Color','g','linewidth',0.5);
lgdhandles = [lgdhandles,hl17];

% left high threshold boundaries
hl18 = line( ...
sdata(pm1).c1lxl+[sdata(pm1).c1profileHighIndL+lineoutAvr,sdata(pm1).c1profileHighIndL+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','b','linewidth',0.5);
hl19 = line( ...
sdata(pm1).c1lxl+[sdata(pm1).c1profileHighIndL-lineoutAvr,sdata(pm1).c1profileHighIndL-lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','b','linewidth',0.5);
hl20 = line( ...
sdata(pm1).c1lxl+[sdata(pm1).c1profileHighIndL-lineoutAvr,sdata(pm1).c1profileHighIndL+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','b','linewidth',0.5);
hl21 = line( ...
sdata(pm1).c1lxl+[sdata(pm1).c1profileHighIndL-lineoutAvr,sdata(pm1).c1profileHighIndL+lineoutAvr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...
'Color','b','linewidth',0.5);
lgdhandles = [lgdhandles,hl21];

% right roi boundaries
hl22 = line( ...
[sdata(pm1).c1rxl,sdata(pm1).c1rxl], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','m','linewidth',1.0);
hl23 = line( ...
[sdata(pm1).c1rxr,sdata(pm1).c1rxr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','m','linewidth',1.0);
hl24 = line( ...
[sdata(pm1).c1rxl,sdata(pm1).c1rxr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','m','linewidth',1.0);
hl25 = line( ...
[sdata(pm1).c1rxl,sdata(pm1).c1rxr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...
'Color','m','linewidth',1.0);

% right low threshold boundaries
hl26 = line( ...
sdata(pm1).c1rxl+[sdata(pm1).c1profileLowIndR+lineoutAvr,sdata(pm1).c1profileLowIndR+lineoutAvr], ...

```

```

[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','r','linewidth',0.5);
hl27 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileLowIndR-lineoutAvr,sdata(pm1).c1profileLowIndR-lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','r','linewidth',0.5);
hl28 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileLowIndR-lineoutAvr,sdata(pm1).c1profileLowIndR+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','r','linewidth',0.5);
hl29 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileLowIndR-lineoutAvr,sdata(pm1).c1profileLowIndR+lineoutAvr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...
'Color','r','linewidth',0.5);

% right mid threshold boundaries
hl30 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileMidIndR+lineoutAvr,sdata(pm1).c1profileMidIndR+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','g','linewidth',0.5);
hl31 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileMidIndR-lineoutAvr,sdata(pm1).c1profileMidIndR-lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','g','linewidth',0.5);
hl32 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileMidIndR-lineoutAvr,sdata(pm1).c1profileMidIndR+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','g','linewidth',0.5);
hl33 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileMidIndR-lineoutAvr,sdata(pm1).c1profileMidIndR+lineoutAvr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...
'Color','g','linewidth',0.5);

% right high threshold boundaries
hl34 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileHighIndR+lineoutAvr,sdata(pm1).c1profileHighIndR+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','b','linewidth',0.5);
hl35 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileHighIndR-lineoutAvr,sdata(pm1).c1profileHighIndR-lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yb], ...
'Color','b','linewidth',0.5);
hl36 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileHighIndR-lineoutAvr,sdata(pm1).c1profileHighIndR+lineoutAvr], ...
[sdata(pm1).c1yt,sdata(pm1).c1yt], ...
'Color','b','linewidth',0.5);
hl37 = line( ...
sdata(pm1).c1rx1+[sdata(pm1).c1profileHighIndR-lineoutAvr,sdata(pm1).c1profileHighIndR+lineoutAvr], ...
[sdata(pm1).c1yb,sdata(pm1).c1yb], ...

```

```

'Color','b','linewidth',0.5);

xLoc = round(linspace(sdata(pm1).c1pxl,sdata(pm1).c1pxr,7));
xVal = (xLoc-sdata(pm1).c1MidDistInd)*params.cam1Scale(logicP1)*1e3;

xValStr = cell(1,length(xVal));
for m=1:length(xVal)
    xValStr{m}=num2str(xVal(m),'%4.2f');
end

set(ha2,'xtick',xLoc);
set(ha2,'xticklabel',xValStr);

yRange = 1:length(image.data(:,1,2));
yUp = floor(length(image.data(:,1,2)).*(params.cam1Scale(logicP1)*1e3));
yLoc = zeros(1,yUp);
for m=1:yUp
    yLoc(m) = find(yRange*(params.cam1Scale(logicP1)*1e3)>=m,1,'first');
end
yVal = yLoc*params.cam1Scale(logicP1)*1e3;

yValStr = cell(1,length(yVal));
for m=1:length(yVal)
    yValStr{m}=num2str(yVal(m),'%2.0f');
end

set(ha2,'ytick',yLoc);
set(ha2,'yticklabel',yValStr);

xlabel('Distance (mm)','fontSize',fontSize);
ylabel('Distance (mm)','fontSize',fontSize);

title( [ '\bf' ...
        'Shot: ' num2str(timap(1,2)) ...
        ' | ' series ' Offset' ...
        ' | +/-' num2str(params.vCharge(logicP)) ' kV' ...
        ' | Camera: ' num2str(timap(1,3)) ...
        ' | Image: ' num2str(timap(1,4)) ...
        '\rm' ],'fontSize',fontSize);

grid off;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);
lgdp = get(hlgd1,'Position');

if sideText==0

```

```

    set(hlgd1,'visible','off')
    lgdp = [lgdp(1),lgdp(2),0,0];
end

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);

set(ht1,'String',infoText);

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2 -a2ptight(1)-txtOff/figOutX,...
        -m3 -a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

if sideText
    set(hlgd1,'position',[...
        a2p(1)+a2p(3)+txtOff/figOutX,...
        a2p(2)+a2p(4)-lgdp(4),...
        lgdp(3),lgdp(4)]);
    lgdp = get(hlgd1,'position');
end

```

```

set(hf1,'position',[...
    lgd(1),...
    lgd(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);
filename = [ ...
    'Shot-' num2str(timap(1,2),'%05.0f) ...
    '_Camera-' num2str(timap(1,3),'%03.0f) ...
    '_Image-' num2str(timap(1,4),'%03.0f) ...
    ];
else
    filename = [ ...
        'Shot-' num2str(timap(1,2),'%05.0f) ...
        '_Camera-' num2str(timap(1,3),'%03.0f) ...
        '_Image-' num2str(timap(1,4),'%03.0f) ...
        '_noText'];
end

%print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);
print(hf1, '-dtiff', ['-r' num2str(300)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'annImages'];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\annImages']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

clear image temp;

end

% Camera 2
logicP1 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP1) > 0
    logic2 = imap(:,3)==params.cam2ID(logicP1);
    logic3 = imap(:,4)==params.cam2PreID(logicP1);
    logic4 = imap(:,4)==params.cam2ImID(logicP1);
    timap = imap(logic1 & logic2 & logic4,:);

    [image,temp] = load_images(timap(1,2:4),imType,shotPath);

```

```

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);

lgdnames = {'Anode','Midplane','Foil Initial Position','ROI', ...
    [num2str(profileLow*100,'%3.0f') '% Position'],'Mean Position', ...
    [num2str(profileHigh*100,'%3.0f') '% Position']};

if sideText
    lgdlogic = true(length(lgdnames),1);
    infoText = { ...
        '\bfShot Parameters\rm',...
        ['Time = ' num2str(sdata(k).cam1Time*1e9,'%4.0f') ' ns'],...
        ['Current = ' num2str(abs(sdata(k).cam1Cur),'%4.0f') ' kA']...
    };
else
    lgdlogic = false(length(lgdnames),1);
    infoText = '';
end

%lgdlogic = true(length(lgdnames),1);
lgdhandles = [];

imagesc(image.data(:,,:2));

```

```

colormap(gray);
hold on;

axis([sdata(pm1).c2pxl-25,sdata(pm1).c2pxr+25,-inf,inf])

% anode boundaries
hl1 = line( ...
    [sdata(pm1).c2pxl,sdata(pm1).c2pxl], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','c','linewidth',2.0);
hl2 = line( ...
    [sdata(pm1).c2pxr,sdata(pm1).c2pxr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','c','linewidth',2.0);
lgdhandles = [lgdhandles,hl2];

% midplane
hl3 = line( ...
    [sdata(pm1).c2MidDistfnd,sdata(pm1).c2MidDistfnd], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','c','linewidth',2.0,'linestyle','--');
lgdhandles = [lgdhandles,hl3];

% initial foil boundaries
hl4 = line( ...
    [sdata(pm1).c2pfxl,sdata(pm1).c2pfxl], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','y','linewidth',1.0);
hl5 = line( ...
    [sdata(pm1).c2pfxr,sdata(pm1).c2pfxr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','y','linewidth',1.0);
lgdhandles = [lgdhandles,hl5];

% left roi boundaries
hl6 = line( ...
    [sdata(pm1).c2lxl,sdata(pm1).c2lxl], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','m','linewidth',1.0);
hl7 = line( ...
    [sdata(pm1).c2lxl,sdata(pm1).c2lxl], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','m','linewidth',1.0);
hl8 = line( ...
    [sdata(pm1).c2lxl,sdata(pm1).c2lxl], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yt], ...
    'Color','m','linewidth',1.0);
hl9 = line( ...
    [sdata(pm1).c2lxl,sdata(pm1).c2lxl], ...

```

```

[sdata(pm1).c2yb,sdata(pm1).c2yb], ...
'Color','m','linewidth',1.0);
lgdhandles = [lgdhandles,h19];

% left low threshold boundaries
h110 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileLowIndL+lineoutAvr,sdata(pm1).c2profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','r','linewidth',0.5);
h111 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileLowIndL-lineoutAvr,sdata(pm1).c2profileLowIndL-lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','r','linewidth',0.5);
h112 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileLowIndL-lineoutAvr,sdata(pm1).c2profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yt], ...
    'Color','r','linewidth',0.5);
h113 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileLowIndL-lineoutAvr,sdata(pm1).c2profileLowIndL+lineoutAvr], ...
    [sdata(pm1).c2yb,sdata(pm1).c2yb], ...
    'Color','r','linewidth',0.5);
lgdhandles = [lgdhandles,h113];

% left mid threshold boundaries
h114 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileMidIndL+lineoutAvr,sdata(pm1).c2profileMidIndL+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);
h115 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileMidIndL-lineoutAvr,sdata(pm1).c2profileMidIndL-lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);
h116 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileMidIndL-lineoutAvr,sdata(pm1).c2profileMidIndL+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yt], ...
    'Color','g','linewidth',0.5);
h117 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileMidIndL-lineoutAvr,sdata(pm1).c2profileMidIndL+lineoutAvr], ...
    [sdata(pm1).c2yb,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);
lgdhandles = [lgdhandles,h117];

% left high threshold boundaries
h118 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileHighIndL+lineoutAvr,sdata(pm1).c2profileHighIndL+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','b','linewidth',0.5);
h119 = line( ...
    sdata(pm1).c2lx1+[sdata(pm1).c2profileHighIndL-lineoutAvr,sdata(pm1).c2profileHighIndL-lineoutAvr], ...

```

```

[sdata(pm1).c2yt,sdata(pm1).c2yb], ...
'Color','b','linewidth',0.5);
hl20 = line( ...
sdata(pm1).c2lxl+[sdata(pm1).c2profileHighIndL-lineoutAvr,sdata(pm1).c2profileHighIndL+lineoutAvr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yt], ...
'Color','b','linewidth',0.5);
hl21 = line( ...
sdata(pm1).c2lxl+[sdata(pm1).c2profileHighIndL-lineoutAvr,sdata(pm1).c2profileHighIndL+lineoutAvr], ...
[sdata(pm1).c2yb,sdata(pm1).c2yb], ...
'Color','b','linewidth',0.5);
lgdhandles = [lgdhandles,hl21];

% right roi boundaries
hl22 = line( ...
[sdata(pm1).c2rxl,sdata(pm1).c2rxl], ...
[sdata(pm1).c2yt,sdata(pm1).c2yb], ...
'Color','m','linewidth',1.0);
hl23 = line( ...
[sdata(pm1).c2rxr,sdata(pm1).c2rxr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yb], ...
'Color','m','linewidth',1.0);
hl24 = line( ...
[sdata(pm1).c2rxl,sdata(pm1).c2rxr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yt], ...
'Color','m','linewidth',1.0);
hl25 = line( ...
[sdata(pm1).c2rxl,sdata(pm1).c2rxr], ...
[sdata(pm1).c2yb,sdata(pm1).c2yb], ...
'Color','m','linewidth',1.0);

% right low threshold boundaries
hl26 = line( ...
sdata(pm1).c2rxl+[sdata(pm1).c2profileLowIndR+lineoutAvr,sdata(pm1).c2profileLowIndR+lineoutAvr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yb], ...
'Color','r','linewidth',0.5);
hl27 = line( ...
sdata(pm1).c2rxl+[sdata(pm1).c2profileLowIndR-lineoutAvr,sdata(pm1).c2profileLowIndR-lineoutAvr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yb], ...
'Color','r','linewidth',0.5);
hl28 = line( ...
sdata(pm1).c2rxl+[sdata(pm1).c2profileLowIndR-lineoutAvr,sdata(pm1).c2profileLowIndR+lineoutAvr], ...
[sdata(pm1).c2yt,sdata(pm1).c2yt], ...
'Color','r','linewidth',0.5);
hl29 = line( ...
sdata(pm1).c2rxl+[sdata(pm1).c2profileLowIndR-lineoutAvr,sdata(pm1).c2profileLowIndR+lineoutAvr], ...
[sdata(pm1).c2yb,sdata(pm1).c2yb], ...
'Color','r','linewidth',0.5);

% right mid threshold boundaries

```

```

hl30 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileMidIndR+lineoutAvr,sdata(pm1).c2profileMidIndR+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);
hl31 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileMidIndR-lineoutAvr,sdata(pm1).c2profileMidIndR-lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);
hl32 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileMidIndR-lineoutAvr,sdata(pm1).c2profileMidIndR+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yt], ...
    'Color','g','linewidth',0.5);
hl33 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileMidIndR-lineoutAvr,sdata(pm1).c2profileMidIndR+lineoutAvr], ...
    [sdata(pm1).c2yb,sdata(pm1).c2yb], ...
    'Color','g','linewidth',0.5);

% right high threshold boundaries
hl34 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileHighIndR+lineoutAvr,sdata(pm1).c2profileHighIndR+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','b','linewidth',0.5);
hl35 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileHighIndR-lineoutAvr,sdata(pm1).c2profileHighIndR-lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yb], ...
    'Color','b','linewidth',0.5);
hl36 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileHighIndR-lineoutAvr,sdata(pm1).c2profileHighIndR+lineoutAvr], ...
    [sdata(pm1).c2yt,sdata(pm1).c2yt], ...
    'Color','b','linewidth',0.5);
hl37 = line( ...
    sdata(pm1).c2rx1+[sdata(pm1).c2profileHighIndR-lineoutAvr,sdata(pm1).c2profileHighIndR+lineoutAvr], ...
    [sdata(pm1).c2yb,sdata(pm1).c2yb], ...
    'Color','b','linewidth',0.5);

xLoc = round(linspace(sdata(pm1).c2pxl,sdata(pm1).c2pxr,7));
xVal = (xLoc-sdata(pm1).c2MidDistInd)*params.cam2Scale(logicP1)*1e3;

xValStr = cell(1,length(xVal));
for m=1:length(xVal)
    xValStr{m}=num2str(xVal(m),'%4.2f');
end

set(ha2,'xtick',xLoc);
set(ha2,'xticklabel',xValStr);

yRange = 1:length(image.data(:,1,2));
yUp = floor(length(image.data(:,1,2)).*(params.cam2Scale(logicP1)*1e3));
yLoc = zeros(1,yUp);

```

```

for m=1:yUp
    yLoc(m) = find(yRange*(params.cam2Scale(logicP1)*1e3)>=m,1,'first');
end
yVal = yLoc*params.cam2Scale(logicP1)*1e3;

yValStr = cell(1,length(yVal));
for m=1:length(yVal)
    yValStr{m}=num2str(yVal(m),%2.0f);
end

set(ha2,'ytick',yLoc);
set(ha2,'yticklabel',yValStr);

xlabel('Distance (mm)','fontSize',fontSize);
ylabel('Distance (mm)','fontSize',fontSize);
title( ['\bf' ...
        'Shot: ' num2str(timap(1,2)) ...
        ' | ' series ' Offset' ...
        ' | +/-' num2str(params.vCharge(logicP)) ' kV' ...
        ' | Camera: ' num2str(timap(1,3)) ...
        ' | Image: ' num2str(timap(1,4)) ...
        '\bf'],'fontSize',fontSize);

grid off;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

if sideText==0
    set(hlgd1,'visible','off')
    lgdp = [lgdp(1),lgdp(2),0,0];
end

ht1 = annotation('textbox',lgdp);

set(ht1,'String',infoText);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);

set(ht1,'FitHeightToText','on');

```

```

set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

if sideText
    set(hlgd1,'position',[...
        a2p(1)+a2p(3)+txtOff/figOutX,...
        a2p(2)+a2p(4)-lgdp(4),...
        lgdp(3),lgdp(4)]);
    lgdp = get(hlgd1,'position');

    set(ht1,'position',[...
        lgdp(1),...
        lgdp(2)-txtOff/figOutY-tp(4),...
        tp(3),...
        tp(4)]);
    filename = [ ...
        'Shot-' num2str(timap(1,2),'%05.0f') ...
        '_Camera-' num2str(timap(1,3),'%03.0f') ...
        '_Image-' num2str(timap(1,4),'%03.0f') ...
        ];
else
    filename = [ ...
        'Shot-' num2str(timap(1,2),'%05.0f') ...
        '_Camera-' num2str(timap(1,3),'%03.0f') ...
        '_Image-' num2str(timap(1,4),'%03.0f') ...
        '_noText'];
end

```

```

end

%print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);
print(hf1, '-dtiff', ['-r' num2str(300)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'annImages');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'annImages']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

clear image temp;

end

end

toc
end
% ===== end =====

% ===== start =====
% Plot Timing Signals
if genTimePlot
display('Plotting Timing Signals');

% Single Shot Timing
if genSingleTime

for k=1:snum
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    hf1=figure('Visible','off');

    a1p = [0,0,1,1];
    a2p = [0,0,1,1];

    set(gcf,'paperunits','inches');

```

```

set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position','a1p,...
    'Visible','off...
);

ha2 = axes(...
    'Position','a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Drive Current','Camera 1','Camera 2','Spec Gate'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hl1=line( sdata(pm1).cTimes*1e9, ...
    sdata(pm1).current, 'color','k','linewidth',2.0);
lgdhandles = [lgdhandles,hl1];

logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
if sum(logicP1) > 0
    hl2=line( sdata(pm1).cam1Time*1e9, ...
        sdata(pm1).cam1Cur, ...
        'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
    line([sdata(pm1).cam1Time,sdata(pm1).cam1Time]*1e9,[0,sdata(pm1).cam1Cur], ...
        'linestyle','-','color','g','linewidth',2.0);
    lgdhandles = [lgdhandles,hl2];
else
    lgdlogic(2)=false;
end

logicP2 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);
if sum(logicP2) > 0
    hl3=line( sdata(pm1).cam2Time*1e9, ...
        sdata(pm1).cam2Cur, ...
        'linestyle','none','marker','s','markeredgecolor','k','markerfacecolor',limeGreen);
    line([sdata(pm1).cam2Time,sdata(pm1).cam2Time]*1e9,[0,sdata(pm1).cam2Cur], ...
        'linestyle','-','color',limeGreen,'linewidth',2.0);
    lgdhandles = [lgdhandles,hl3];

```

```

else
    lgdlogic(3)=false;
end

logicP3 = logical(logicP.*params.specOn(logicP));
if sum(logicP3) > 0 && plotSpec
    fytick = get(ha2,'ytick');
    hl4=line([sdata(pm1).specTime,sdata(pm1).specTime]*1e9, ...
            [0,fytick(end)], ...
            'linestyle','-','color','c','linewidth',1.0);
    hl5 = line([sdata(pm1).specTime,sdata(pm1).specTime]+params.specGate(logicP)*1e9, ...
            [0,fytick(end)], ...
            'linestyle','-','color','c','linewidth',1.0);
    hr1 = rectangle('position',[sdata(pm1).specTime*1e9,0,params.specGate(logicP)*1e9,fytick(end)], ...
            'facecolor',lightCyan,'linestyle','none');
    uistack(hl4,'bottom')
    uistack(hl5,'bottom')
    uistack(hr1,'bottom')
    lgdhandles = [lgdhandles,hl4];

    filename = [ ...
        'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
        '_TimePlot'];

else
    lgdlogic(4)=false;

    filename = [ ...
        'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
        '_TimePlot_noSpec'];

end

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

yUp = ceil(sdata(pm1).cMax/100)*100;

set(ha2,'xlim',[-50,350]);
set(ha2,'ylim',[0,yUp]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfCurrent (kA)\rm','fontSize',fontSize);

```

```

title(['\bfTiming Signals | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/-' num2str(params.vCharge(logicP)) '
kV\rm'], 'fontSize',fontSize);

```

```

grid on;

```

```

box on;

```

```

fytick = get(ha2,'ytick');

```

```

fytickstr = cell(1,length(fytick));

```

```

for j=1:length(fytick)

```

```

    fytickstr{j}=num2str(fytick(j), '%4.0f');

```

```

end

```

```

set(ha2,'yticklabel',fytickstr);

```

```

set(hf1,'CurrentAxes',ha1);

```

```

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

```

```

lgdp = get(hlgd1,'Position');

```

```

ht1 = annotation('textbox',lgdp);

```

```

set(ht1,'HorizontalAlignment','Left');

```

```

set(ht1,'VerticalAlignment','Top');

```

```

set(ht1,'Margin',0);

```

```

set(ht1,'LineStyle','none');

```

```

set(ht1,'FontSize',fontSize);

```

```

set(ht1,'String',...

```

```

    { ...

```

```

    '\bfShot Parameters\rm',...

```

```

    ['Peak Current = ' num2str(abs(sdata(k).cMax), '%4.0f') ' kA'],...

```

```

    ['Peak Time = ' num2str(sdata(k).cMaxT*1e9, '%3.0f') ' ns'],...

```

```

    });

```

```

set(ht1,'FitHeightToText','on');

```

```

set(ht1,'FitBoxToText','on');

```

```

tp = get(ht1,'position');

```

```

a2ptight = get(ha2,'tightinset');

```

```

a2pout = get(ha2,'outerposition');

```

```

m1 = tp(3) + 2*txtOff/figOutX;

```

```

m2 = lgdp(3) + 2*txtOff/figOutX;

```

```

m3 = txtOff/figOutY + a2ptight(4);

```

```

if ( m1 > m2 )

```

```

    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...

```

```

        a2ptight(2)+txtOff/figOutY,...

```

```

        -m1 -a2ptight(1)-txtOff/figOutX,...

```

```

        -m3-a2ptight(2)-txtOff/figOutY...
    );
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
    ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)];
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\'], 'timePlots');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath 'timePlots']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

end

% Image Timing Summary
if genSumTime

hf1=figure('Visible','off');

alp = [0,0,1,1];

```

```

a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize',[figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Drive Current','Camera 1','Camera 2'};
lgdlogic = false(length(lgdnames),1);

maxC = 0;

for k=1:snnum
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);

    hl1=line( sdata(pm1).cTimes*1e9, ...
        sdata(pm1).current, 'color','k','linewidth',1.0);
    lgdlogic(1) = true;

    logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
    if sum(logicP1) > 0
        hl2=line( sdata(pm1).cam1Time*1e9, ...
            sdata(pm1).cam1Cur, ...
            'linestyle','none','marker','o','markeredgecolor','k','markerfacecolor','g');
        line([sdata(pm1).cam1Time,sdata(pm1).cam1Time]*1e9,[0,sdata(pm1).cam1Cur], ...
            'linestyle','-','color','g');
        lgdlogic(2) = true;
    end

    logicP2 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);

```

```

if sum(logicP2) > 0
    hl3=line( sdata(pm1).cam2Time*1e9, ...
            sdata(pm1).cam2Cur, ...
            'linestyle','none','marker','s','markeredgecolor','k','markerfacecolor',limeGreen);
    line([sdata(pm1).cam2Time,sdata(pm1).cam2Time]*1e9,[0,sdata(pm1).cam2Cur], ...
        'linestyle','--','color',limeGreen);
    lgdlogic(3) = true;
end

maxC = max(maxC,sdata(pm1).cMax);
end

lgdhandles = [];
for k=1:length(lgdlogic)
    if lgdlogic(k)
        lgdhandles = [lgdhandles,eval(['hl' num2str(k)])];
    end
end

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%3.0f');
end
set(ha2,'yticklabel',fytickstr);

yUp = ceil(maxC/100)*100;

set(ha2,'xlim',[-50,350]);
set(ha2,'ylim',[0,yUp]);

xlabel('\bfTime (ns)\rm','fontsize',fontSize);
ylabel('\bfCurrent (kA)\rm','fontsize',fontSize);

title(['\bfImage Timing | ' series ' Offset\rm'],'fontsize',fontSize);

grid on;
box on;

fytick = get(ha2,'ytick');
fytickstr = cell(1,length(fytick));
for j=1:length(fytick)
    fytickstr{j}=num2str(fytick(j),'%4.0f');
end
set(ha2,'yticklabel',fytickstr);

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

```

```

lgdp = get(hlgd1, 'Position');

ht1 = annotation('textbox',lgdp);

set(ht1, 'HorizontalAlignment', 'Left');
set(ht1, 'VerticalAlignment', 'Top');
set(ht1, 'Margin', 0);
set(ht1, 'LineStyle', 'none');

set(ht1, 'FontSize', fontSize);
set(ht1, 'String', ...
    { ...
    " ...
    });

set(ht1, 'FitHeightToText', 'on');
set(ht1, 'FitBoxToText', 'on');

tp = get(ht1, 'position');

a2ptight = get(ha2, 'tightinset');
a2pout = get(ha2, 'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2, 'position', a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2, 'position');
else
    set(ha2, 'position', a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2, 'position');
end

set(hlgd1, 'position', [...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1, 'position');

```

```

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ timeFile '_Series-' series];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', timeFile];
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\ timeFile]);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

toc
end
% ===== end =====

% ===== start =====
% Plot Plasma Dynamics
if genDynamicPlots
display('Plotting Plasma Dynamics');

for k=1:snun
    pm1 = pmap(k,1);
    logicP = params.shot==shots(pm1);
    logic1 = imap(:,2)==params.shot(logicP);

    logicP1 = logical(logicP.*params.cam1(logicP).*params.cam1Pre);
    logicP2 = logical(logicP.*params.cam2(logicP).*params.cam2Pre);

    ky = 2*pi/wvlengthY;
    kz = 2*pi/wvlengthZ;
    kx = sqrt(kz^2+ky^2);
    w = lfoil;
    t = sdata(pm1).cTimes;
    cl = sdata(pm1).current*1e3;

```

```

tStart = find(t*1e9>=0,1,'first');
tEnd = find(t*1e9>=350,1,'first');

x = zeros(length(t),1);
v = zeros(length(t),1);
a = zeros(length(t),1);
B1 = zeros(length(t),1);
B3 = zeros(length(t),1);
grOld = zeros(length(t),1);
grNew = zeros(length(t),1);

sLabel = 'Measured';

switch 1
case sum(logicP1)*sum(logicP2)
    d = (((sdata(pm1).c1pxl+sdata(pm1).c1pxr)/2-sdata(pm1).c1pxl)*params.cam1Scale(logicP)+...
        ((sdata(pm1).c2pxl+sdata(pm1).c2pxr)/2-sdata(pm1).c2pxl)*params.cam2Scale(logicP))/2;
    x(1) = (sdata(pm1).c1InitialPos+sdata(pm1).c2InitialPos)/2;
case sum(logicP1)
    d = ((sdata(pm1).c1pxl+sdata(pm1).c1pxr)/2-sdata(pm1).c1pxl)*params.cam1Scale(logicP);
    x(1) = sdata(pm1).c1InitialPos;
case sum(logicP2)
    d = ((sdata(pm1).c2pxl+sdata(pm1).c2pxr)/2-sdata(pm1).c2pxl)*params.cam2Scale(logicP);
    x(1) = sdata(pm1).c2InitialPos;
otherwise
    d = d0;
    x(1) = x0;
    sLabel = 'Intended';
end

v(1) = v0;
a(1) = -2*mu*cl(1)^2/(2*w^2*rho*th*(2*d-th))*x(1);
B1(1) = -mu*cl(1)/w*((d-x(1)-th/2)/(2*d-th));
B3(1) = mu*cl(1)/w*((d+x(1)-th/2)/(2*d-th));
vL = B1(1)/sqrt(mu*rho);
vU = B3(1)/sqrt(mu*rho);
L = kz^2*vL^2;
U = kz^2*vU^2;
grOld(1) = sqrt(kx*a(1));
grNew(1) = sqrt( (L+U)*coth(th*kx)/2 - 0.5*sqrt((L+U)^2*coth(th*kx)^2-4*(U+a(1)*kx)*(L-a(1)*kx)) );

for m = 2:length(t)
    tq = (t(m)-t(m-1))/4;

    xq1 = v(m-1)*tq+x(m-1);
    vq1 = a(m-1)*tq+v(m-1);
    cl1 = ((cl(m)-cl(m-1))/(t(m)-t(m-1)))*1*tq+cl(m-1);
    aq1 = -2*mu*(cl1)^2/(2*w^2*rho*th*(2*d-th))*xq1;

```

```

xq2 = vq1*tq+xq1;
vq2 = aq1*tq+vq1;
cl2 = ((cl(m)-cl(m-1))/(t(m)-t(m-1)))^2*tq+cl(m-1);
aq2 = -2*mu*(cl2)^2/(2*w^2*rho*th*(2*d-th))*xq2;

xq3 = vq2*tq+xq2;
vq3 = aq2*tq+vq2;
cl3 = ((cl(m)-cl(m-1))/(t(m)-t(m-1)))^3*tq+cl(m-1);
aq3 = -2*mu*(cl3)^2/(2*w^2*rho*th*(2*d-th))*xq3;

xm = vq3*tq+xq3;
vm = aq3*tq+vq3;
cl4 = ((cl(m)-cl(m-1))/(t(m)-t(m-1)))^4*tq+cl(m-1);
am = -2*mu*(cl4)^2/(2*w^2*rho*th*(2*d-th))*xm);

B1(m) = -mu*cl(m)/w*((d-x(m)-th/2)/(2*d-th));
B3(m) = mu*cl(m)/w*((d+x(m)-th/2)/(2*d-th));
vL = B1(m)/sqrt(mu*rho);
vU = B3(m)/sqrt(mu*rho);
L = kz^2*vL^2;
U = kz^2*vU^2;

grOld(m) = sqrt(kx*a(m));
grNew(m) = sqrt( (L+U)*coth(th*kx)/2 - 0.5*sqrt((L+U)^2*coth(th*kx)^2-4*(U+a(m)*kx)*(L-a(m)*kx)) );
end

tgrOld = calc_int(t,abs(grOld),1);
tgrNew = calc_int(t,imag(grNew),1);

% kinematics
if genKinematics

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off...

```

```

);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'Position','Velocity','Acceleration',...
    'C1 Left Position','C1 Right Position',...
    'C2 Left Position','C2 Right Position'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on

hl1=line(t*1e9,x*1e3, ...
    'color','b','linewidth',2.0,'linestyle','--');
lgdhandles = [lgdhandles,hl1];
hl2=line(t*1e9,v*1e-4, ...
    'color','g','linewidth',2.0,'linestyle','--');
lgdhandles = [lgdhandles,hl2];
hl3=line(t*1e9,a*1e-11, ...
    'color','r','linewidth',2.0,'linestyle',':');
lgdhandles = [lgdhandles,hl3];

if sum(logicP1)
    hl4=errorbar( sdata(pm1).cam1Time*1e9,sdata(pm1).c1MidL*1e3,sdata(pm1).c1MidUncL*1e3,'r','linestyle','none', ...
        'marker','s','markeredgecolor','k','markerfacecolor','c');
    lgdhandles = [lgdhandles,hl4];
    hl5=errorbar( sdata(pm1).cam1Time*1e9,sdata(pm1).c1MidR*1e3,sdata(pm1).c1MidUncR*1e3,'g','linestyle','none', ...
        'marker','o','markeredgecolor','k','markerfacecolor','m');
    lgdhandles = [lgdhandles,hl5];
else
    lgdlogic(4)=false;
    lgdlogic(5)=false;
end

if sum(logicP2)
    hl6=errorbar( sdata(pm1).cam2Time*1e9,sdata(pm1).c2MidL*1e3,sdata(pm1).c2MidUncL*1e3,'r','linestyle','none', ...
        'marker','d','markeredgecolor','k','markerfacecolor','y');
    lgdhandles = [lgdhandles,hl6];
    hl7=errorbar( sdata(pm1).cam2Time*1e9,sdata(pm1).c2MidR*1e3,sdata(pm1).c2MidUncR*1e3,'g','linestyle','none', ...
        'marker','^','markeredgecolor','k','markerfacecolor','orangeRed');
    lgdhandles = [lgdhandles,hl7];
else

```

```

    lgdlogic(6)=false;
    lgdlogic(7)=false;
end

hold off

set(ha2,'xlim',[0,350]);
set(ha2,'ylim',[-15,10]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfPosition (mm) | Velocity (cm/\mus) | (1/10) x Acceleration (cm/{\mus}^2)\rm','fontSize',fontSize);

title(['\bfFoil Kinematics | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm', ...
    ['Peak Current = ' num2str(abs(sdata(k).cMax),%4.0f) ' kA'],...
    ['Peak Time = ' num2str(sdata(k).cMaxT*1e9,%4.0f) ' ns'],...
    ", ...
    '\bfAblationless Parameters\rm', ...
    ['L = ' num2str(w*1e3,%3.1f) ' (mm)'], ...
    ['w = ' num2str(th*1e9,%4.0f) ' (nm)'], ...
    ['\rho = ' num2str(rho*1e-3,%4.2f) ' (g/cc)], ...
    ", ...
    ['\bf sLabel ' Quantities\rm'], ...
    ['D = ' num2str(d(1)*1e3,%4.2f) ' (mm)'], ...
    ['X_{0} = ' num2str(x(1)*1e3,%5.2f) ' (mm)], ...
    });

set(ht1,'FitHeightToText','on');

```

```

set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_Kinematics'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'dynamicsKM']);
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\dynamicsKM']);

```

```

else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

% magnetic fields
if genMagFields

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize', [figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position',...
    'XMinorTick','on',...
    'YMinorTick','on',...
    'FontSize',fontSize,...
    'color','w...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'B_{1}','B_{3}','B_{total}'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];
hold on

h11=line(*1e9,B1, ...
    'color','b','linewidth',2.0,'linestyle','-');
lgdhandles = [lgdhandles,h11];

```

```

hl2=line(*1e9,B3, ...
    'color','r','linewidth',2.0,'linestyle','--');
lgdhandles = [lgdhandles,hl2];
hl3=line(*1e9,B1+B3, ...
    'color','k','linewidth',2.0,'linestyle','-');
lgdhandles = [lgdhandles,hl3];

hold off

set(ha2,'xlim',[0,350]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bfMagnetic Field (T)\rm','fontSize',fontSize);

title(['\bfMagnetic Fields | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/-' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
    { ...
    '\bfShot Parameters\rm', ...
    ['Peak Current = ' num2str(abs(sdata(k).cMax),'%4.0f') ' kA'],...
    ['Peak Time = ' num2str(sdata(k).cMaxT*1e9,'%4.0f') ' ns']...
    });

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

```

```

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_MagFields'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'dynamicsBF']);
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\dynamicsBF']);
else
    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

```

```

end

% growth rates
if genGrowthRates

hf1=figure('Visible','off');

a1p = [0,0,1,1];
a2p = [0,0,1,1];

set(gcf,'paperunits','inches');
set(gcf,'paperposition',[0,0,figOutX,figOutY]);
set(gcf,'paperpositionmode','auto');
set(gcf,'PaperSize',[figOutX,figOutY]);
set(gcf,'units','inches');
set(gcf,'position',[1/3,1/3,figOutX,figOutY]);
set(gcf,'units','normalized');

ha1 = axes(...
    'Position',a1p,...
    'Visible','off'...
);

ha2 = axes(...
    'Position',a2p,...
    'activepositionproperty','position'...
    'XMinorTick','on'...
    'YMinorTick','on'...
    'FontSize',fontSize,...
    'color','w'...
);

set(hf1,'CurrentAxes',ha2);
lgdnames = {'gamma Rayleigh','\Gamma Rayleigh','\gamma 3 Region','\Gamma 3 Region'};
lgdlogic = true(length(lgdnames),1);
lgdhandles = [];

hl1=line( t*1e9, abs(grOld)*1e-6, ...
    'color','k','linewidth',2.0);
    lgdhandles = [lgdhandles,hl1];
hl2=line( t*1e9, tgrOld, ...
    'color','k','linewidth',2.0,'linestyle','--');
    lgdhandles = [lgdhandles,hl2];
hl3=line( t*1e9, imag(grNew)*1e-6, ...
    'color','r','linewidth',1.0);
    lgdhandles = [lgdhandles,hl3];
hl4=line( t*1e9, tgrNew, ...
    'color','r','linewidth',1.0,'linestyle','--');

```

```

lgdhandles = [lgdhandles,h14];

set(ha2,'xlim',[0,350]);
set(ha2,'ylim',[0,inf]);

xlabel('\bfTime (ns)\rm','fontSize',fontSize);
ylabel('\bf\gamma (\mus^{-1}) | \Gamma\rm','fontSize',fontSize)

title(['\bfGrowth Rates | ' series ' Offset | Shot ' num2str(pmap(k,2)) ' | +/- ' num2str(params.vCharge(logicP)) '
kV\rm'],'fontSize',fontSize);

grid on;
box on;

set(hf1,'CurrentAxes',ha1);

hlgd1=legend(lgdhandles,lgdnames(lgdlogic),'FontSize',fontSize);

lgdp = get(hlgd1,'Position');

ht1 = annotation('textbox',lgdp);

set(ht1,'HorizontalAlignment','Left');
set(ht1,'VerticalAlignment','Top');
set(ht1,'Margin',0);
set(ht1,'LineStyle','none');

eFoldOld = 1/(mean(abs(grOld(tStart:tEnd)))*1e-9);
eFoldNew = 1/(mean(imag(grNew(tStart:tEnd)))*1e-9);

set(ht1,'FontSize',fontSize);
set(ht1,'String',...
{...
'\bfShot Parameters\rm', ...
['Peak Current = ' num2str(abs(sdata(k).cMax),%4.0f) ' kA'],...
['Peak Time = ' num2str(sdata(k).cMaxT*1e9,%4.0f) ' ns'],...
", ...
'\bfSpatial Frequencies\rm', ...
['k_{x} = ' num2str(kx*1e-3,%4.2f) ' (rad/mm)'], ...
['k_{y} = ' num2str(ky*1e-3,%4.2f) ' (rad/mm)'], ...
['k_{z} = ' num2str(kz*1e-3,%4.2f) ' (rad/mm)'], ...
", ...
'\bfAverage e-Folding Times\rm', ...
['<1/\gamma> Rayleigh = ' num2str(eFoldOld,%5.0f) ' (ns)'], ...
['<1/\gamma> 3 Region = ' num2str(eFoldNew,%5.0f) ' (ns)'], ...
});

set(ht1,'FitHeightToText','on');
set(ht1,'FitBoxToText','on');

```

```

tp = get(ht1,'position');

a2ptight = get(ha2,'tightinset');
a2pout = get(ha2,'outerposition');

m1 = tp(3) + 2*txtOff/figOutX;
m2 = lgdp(3) + 2*txtOff/figOutX;
m3 = txtOff/figOutY + a2ptight(4);
if ( m1 > m2 )
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m1-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
else
    set(ha2,'position',a2p+[a2ptight(1)+txtOff/figOutX,...
        a2ptight(2)+txtOff/figOutY,...
        -m2-a2ptight(1)-txtOff/figOutX,...
        -m3-a2ptight(2)-txtOff/figOutY...
        ]);
    a2p = get(ha2,'position');
end

set(hlgd1,'position',[...
    a2p(1)+a2p(3)+txtOff/figOutX,...
    a2p(2)+a2p(4)-lgdp(4),...
    lgdp(3),lgdp(4)]);
lgdp = get(hlgd1,'position');

set(ht1,'position',[...
    lgdp(1),...
    lgdp(2)-txtOff/figOutY-tp(4),...
    tp(3),...
    tp(4)]);

filename = [ ...
    'Series-' series '_Shot-' num2str(pmap(k,2),'%05.0f') ...
    '_GrowthRates'];

print(hf1, '-dtiff', ['-r' num2str(pltres)], filename);

[s,mess,messID] = mkdir( ...
    [sumPath '\', 'dynamicsGR');
if s==1
    movefile( [filename '.tif'], ...
        [sumPath '\dynamicsGR']);
else

```

```

    display( mess );
    display( num2str(pmap(k,2)) );
    break;
end

close(hf1);

end

end

toc
end
% ===== end =====

if clearOnExit
    clear;
end
display('*** Analysis Completed ***');
toc

```

B.3 Load Traces

```

function [out,idMap]=load_traces(id,delim,location)
% [out,idMap]=load_traces(id,delim,location)
% Loads timing signal traces for the given n shots
% out: 'out' is a structure:
%   inputs: (shot,scope,channel)
%   fields: .times: the time values from the trace
%           .signal: the signal values from the trace
% out: 'idMap' is the nx4 matrix of trace IDs
%   rows: Each row is a trace ID for the output structure
%   cols: 1: index for output structure (int)
%         2: shot ID (int)
%         3: scope ID (int)
%         3: channel ID (int)
% in: 'id' is an nx3 matrix of trace IDs
%   rows: Each row is a trace ID to load
%   cols: 1: shot ID (int)
%         2: oscilloscope ID (int)
%         2: channel ID (int)
% in: 'delim' is the delimiter for the files (string)
% in: 'location' is an n element list of paths to the trace files (string)
% Jacob Zier: 04/09/2008

num = length(id(:,1));

```

```

out(1:num) = struct('times',[], 'signal',[]);
idMap = [(1:num)',id];

for k=1:num
    fileName = dir([location '\ num2str(id(k,1), '%05.0f') \traces\ num2str(id(k,1), '%05.0f') '_' num2str(id(k,2), '%03.0f') '_'
num2str(id(k,3), '%03.0f') '_'*.*)];
    fid = fopen([location '\ num2str(id(k,1), '%05.0f') \traces\ fileName.name], 'r');
    if fid==-1
        fclose('all')
        display(['Error opening file ' num2str(id(k,1), '%05.0f') '_' num2str(id(k,2), '%03.0f') '_' num2str(id(k,3), '%03.0f') '_'*.*)]);
        return;
    else
        data = textscan(fid, '%f %f', 'delimiter', delim);
        out(k).times=data{1};
        out(k).signal=data{2};
    end
    fclose(fid);
end
end
end

```

B.4 Load Parameters (MRT)

```

function [out,idMap]=load_paramSingle(id,param,location)
% [out,idMap]=load_paramSingle(id,param,location)
% Loads shot parameters from a CSV file for the given
% shot and parameter
% out: 'out' is a structure:
% inputs: (1)
% fields: determined by the header line in the csv file
% out: 'idMap' is the nx2 matrix of parameter IDs
% rows: Each row is shot parameter ID for the output structure
% cols: 1: index for output structure (int)
%      2: shot id (int)
% in: 'id' is an nx1 matrix of shot IDs
% rows: Each row is a shot to load
% cols: 1: shot number
% in: 'param' is the parameter type to load (string)
% in: 'location' is an n element list of paths to the
%      parameter files (cell array of strings)
% Jacob Zier: 04/09/2008

num = length(id(:,1));
idMap = [(1:num)',id];
fileName = dir([location '\ param '.csv]);
fid = fopen([location '\ fileName.name], 'r');
if fid==-1
    fclose('all')
    display(['Error opening file ' param '.csv']);

```

```

return
else

rawFields = textscan(fid, '%s', 1, 'delimiter', '');
colNum = length(strfind(char(rawFields{1}),','))+1;
fieldFormats = '';
for i=1:colNum
    fieldFormats = [fieldFormats, '%s'];
end
rowDataFormats = textscan(fid, fieldFormats, 1, 'delimiter', ',');
dataFormats = '';
for i=1:colNum
    dataFormats = [dataFormats, char(rowDataFormats{i})];
end
fields = textscan(char(rawFields{1}), fieldFormats, 'delimiter', ',');
data = textscan(fid, dataFormats, 'delimiter', ',');
for i=1:length(fields(1,:))
    out.(char(fields{i}))=data{i};
end
fclose(fid);
end
end

```

B.5 Load Parameters (Correlated Ablation)

```

function [out,idMap]=load_param(id,param,location)
% [out,idMap]=load_param(id,param,location)
% Loads shot parameters from a CSV file for the given
% shot and parameter
% out: 'out' is a structure:
% inputs: (id)
% fields: determined by the header line in the csv file
% out: 'idMap' is the nx2 matrix of parameter IDs
% rows: Each row is shot parameter ID for the output structure
% cols: 1: index for output structure (int)
%      2: shot id (int)
% in: 'id' is an nx1 matrix of shot IDs
% rows: Each row is a shot to load
% cols: 1: shot number
% in: 'param' is the parameter type to load (string)
% in: 'location' is an n element list of paths to the
%      parameter files (cell array of strings)
% Jacob Zier: 04/09/2008

num = length(id(:,1));
idMap = [(1:num)',id];
for k=1:num
    fileName = dir([location '\ num2str(id(k,1),'%05.0f') '\ num2str(id(k,1),'%05.0f') '_' param '.csv']);

```

```

fid = fopen([location '\ num2str(id(k,1),'%05.0f') '\ fileName.name'],'r');

if fid==-1
    fclose('all')
    display(['Error opening file ' num2str(id(k,1),'%05.0f') ' _ ' param '.csv']);
    return
else
    rawFields = textscan(fid, '%s', 1, 'delimiter', '');
    colNum = length(strfind(char(rawFields{1}),','))+1;
    fieldFormats = "";
    for i=1:colNum
        fieldFormats = [fieldFormats, '%s'];
    end
    rawDataFormats = textscan(fid, fieldFormats, 1, 'delimiter', '');
    dataFormats = "";
    for i=1:colNum
        dataFormats = [dataFormats, char(rawDataFormats{i})];
    end
    fields = textscan(char(rawFields{1}), fieldFormats, 'delimiter', '');
    data = textscan(fid, dataFormats, 'delimiter', '');
    for i=1:length(fields(1,:))
        out(k).(char(fields{i}))=data{i};
    end
    fclose(fid);
end
end
end

```

B.6 Load Images

```

function [out,idMap]=load_images(id,format,location)
% [out,idMap]=load_images(id,format,location)
% Loads images for the given shot
% out: 'out' is a structure:
%   fields: .data   image matrix converted to 'double'
%           .raw   image matrix
%           .map   image color map
%           .format the format for the original image
% out: 'idMap' is the nx3 matrix of image IDs
%   rows: Each row is an image ID for the output structure
%   cols: 1: index for output structure (int)
%         2: shot id (int)
%         3: image id (int)
% in: 'id' is the nx2 or nx3 matrix of image IDs
%   rows: Each row is an image ID to load
%   cols: 1: shot ID (int)
%         2: image ID (int)
% in: 'format' is the file extension (string)

```

```

% in: 'location' is an n element list of paths to the image files (string)
% Jacob Zier: 04/09/2008

num = length(id(:,1));
out(1:num) = struct('data',0,'raw',0,'map',0,'format','');
idMap = [(1:num)',id];
for k=1:num
    if length(id)==2
        fileName = dir([location '\ num2str(id(k,1),%05.0f) \images\ num2str(id(k,1),%05.0f) '_' num2str(id(k,2),%03.0f) '_' *.'
format]);
    elseif length(id)==3
        fileName = dir([location '\ num2str(id(k,1),%05.0f) \images\ num2str(id(k,1),%05.0f) '_' num2str(id(k,2),%03.0f) '_' '
num2str(id(k,3),%03.0f) '_' *.' format]);
    end
    if isempty(fileName)
        display(['Error opening file ' num2str(id(k,1),%05.0f) '_' num2str(id(k,2),%03.0f) '_' *.' format]);
        return;
    else
        [pic,map] = imread([location '\ num2str(id(k,1),%05.0f) \images\ fileName.name]);
        out(k).data=double(pic);
        out(k).raw=pic;
        out(k).map=map;
        out(k).format=format;
    end
end
end
end

```

B.7 Find Wire Lineout

```

function [wire,wirestd,back,backstd,nx,ycoords] = find_wireLineout(im,x1,y1,x2,y2,h,wh)
% [wire,wirestd,back,backstd,nx,ycoords] = ...
% find_wireLineout(image,x1,y1,x2,y2,h,wh);
% Generates lineouts of horizontal wires from a greyscale radiograph
% out: 'wire' contains the raw wire lineout
% out: 'wirestd' contains the wire standard deviation lineout
% out: 'back' contains the raw local background lineout
% out: 'backstd' contains the background standard dev. lineout
% out: 'nx' contains the horizontal wire coordinates
% out: 'ycoords' contains wh x nx vertical wire coordinates
% in: 'im' is the radiograph (negative) with a horizontal wire
% in: (x1,y1) is the left wire endpoint in pixels
% in: (x2,y2) is the right wire endpoint in pixels
% in: 'h' is the vertical radius bounding the wire in pixels
% in: 'wh' is the vertical thickness of the wire in pixels
% Jacob Zier: 04/24/2008

nx = linspace(x1,x2,x2-x1+1);
ny = round(y1+((y2-y1)/(x2-x1)).*(nx-x1));
wire = zeros(length(nx),1);

```

```

wirestd = zeros(length(nx),1);
back = zeros(length(nx),1);
backstd = zeros(length(nx),1);
field = zeros(2*h+1,length(nx));
sfield = zeros(2*h+1,length(nx));
ycoords = zeros(wh,length(nx));
wh2 = ceil(wh/2);
for i=1:length(nx)
    field(:,i) = im(ny(i)-h:ny(i)+h,nx(i));
    sfield(:,i) = sort(field(:,i),'descend');
    % wire(i) = mean(sfield(1:wh,i));
    wire(i) = mean(sfield(1:wh2,i));
    wirestd(i) = std(sfield(1:wh2,i));
    back(i) = mean(sfield(wh:end,i));
    backstd(i) = std(sfield(wh:end,i));
    j=1;
    while j<=wh2
        sfcoords = find(im(:,nx(i))==sfield(j,i));
        sfrcoords = sfcoords( sfrcoords <= ny(i)+h & sfrcoords >= ny(i)-h );
        for k=1:length(sfrcoords)
            ycoords(j,i) = sfrcoords(k);
            j=j+1;
            if j>wh2
                break
            end
        end
    end
end
end
end

```

B.8 Generate Pairs

```

function [pairs]=genPairs(magData,pairPer)
% [pairs]=genPairs(magData,pairPer)
% Calculates wire ID pairs based on wire magnification
% out: 'pairs' is a nx2 matrix of unique wire ID pairs:
% in: 'magData' is the nx2 matrix of wire magnifications
%   rows: Each row is a wire
%   cols: 1: wire ID (int)
%         2: magnification (double)
% in: 'pairPer' is the allowed percent difference in magnification (double)
% Jacob Zier: 01/15/2008

magData = unique(magData,'rows');
pairs = [];
for k=1:length(magData(:,1))
    logic1 = abs((magData(:,2)-magData(k,2))./magData(k,2))*100<=pairPer;
    logic1(k) = 0;

```

```

pairs = [pairs;ones(sum(logic1),1).*magData(k,1),magData(logic1,1)];
end

for k=1:length(pairs(:,1))
    if pairs(k,1)~= -1;
        logic1 = pairs(:,1)==pairs(k,2);
        logic1(k) = 0;
        logic2 = pairs(:,2)==pairs(k,1);
        logic2(k) = 0;
        pairs(logic1 & logic2,1) = -1;
    end
end
logic1 = pairs(:,1)~= -1;
pairs = [pairs(logic1,1),pairs(logic1,2)];
end

```

B.9 Integrate with Array Reduction

```

function [f] = calc_int(tarray,dfdt,index)
% [f] = calc_int(tarray,signal,index)
% Integrates a given signal.
% out: 'f' is the integrated signal.
% in: 'tarray' is the given time array.
% in: 'dfdt' is the given time derivative signal array.
% in: 'index' is the index to begin integration.
% Jacob Zier: 5/01/2008

f = zeros(1,length(dfdt(index:end-1)));
if index==1
    bias = 0;
else
    bias = mean(dfdt(1:index-1));
end
if (length(tarray)>1)
    f = cumtrapz(tarray(index:end),dfdt(index:end)-bias);
else
    f(1)=dfdt(1).*tarray(1);
end
end

```

B.10 Integrate with Leading Zeros

```

function [f] = calc_intz(tarray,dfdt,index)
% [f] = calc_int(tarray,signal,index)
% Integrates a given signal.
% out: 'f' is the integrated signal with leading zeros.
% in: 'tarray' is the given time array.

```

```

% in: 'dfdt' is the given time derivative signal array.
% in: 'index' is the index to begin integration.
% Jacob Zier: 03/30/2010

%f = zeros(1,length(dfdt(index:end-1)));
f = zeros(1, length(dfdt));
if index==1
    bias = 0;
else
    bias = mean(dfdt(1:index-1));
end
if (length(tarray)>1)
    f(index:end) = cumtrapz(tarray(index:end),dfdt(index:end)-bias);
else
    f(1)=dfdt(1).*tarray(1);
end
end

```

B.11 FFT 1D

```

function [mag,arg,ksx] = fft1map(data,distx)
% [mag,arg,ksx] = fft1map(data,distx)
% Analyzes a 1-D profile in k-space.
% Jacob Zier: 03/04/2009

resx = length(data);
fx = resx/distx;
nyquistx = fx/2;
fftresx = resx;
ffts = fft(data);
mag = abs(ffts(1:floor(fftresx/2)));
arg = angle(ffts(1:floor(fftresx/2)));
ksx = linspace(0,2*pi*nyquistx,length(mag));
end

```

B.12 Find Peak Time

```

function [tout,sindex] = find_peakTime(tarray,signal,toffset)
% [tout,index] = find_peaktime(tarray,signal,offset)
% Finds the time where the maximum occurs in a given signal.
% out: 'tout' is the time of the given signal.
% out: 'sindex' is the index of the signal array for TOUT.
% in: 'tarray' is the given time array.
% in: 'signal' is the given signal array with a peak.
% in: 'toffset' is the output time shift.
% Jacob Zier: 12/15/2008

```

```
[maxsig,sindex]=max(signal);  
tout=tarray(sindex)-toffset;  
end
```

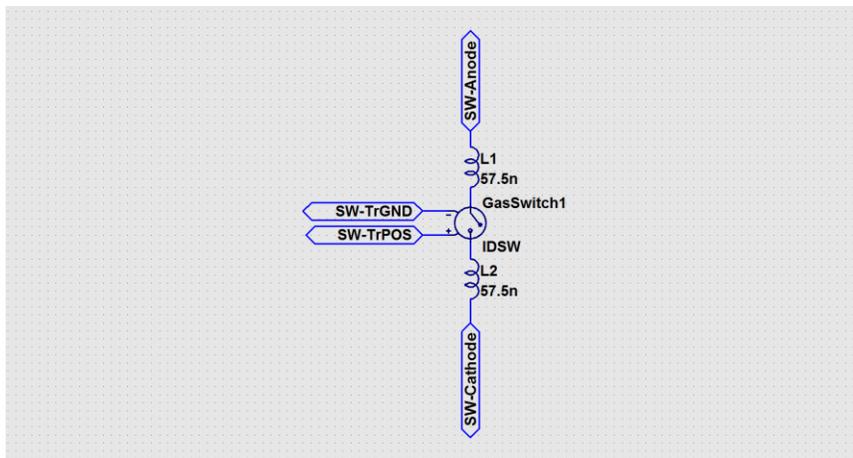
APPENDIX C

CIRCUIT MODEL

C.1 Switch

Netlist

```
.subckt switch_hcei SW-Anode SW-Cathode SW-TrGND SW-TrPOS
L1 SW-Anode N001 57.5n Rser=30m Rpar=0 Cpar=0
L2 N002 SW-Cathode 57.5n Rser=30m Rpar=0 Cpar=0
S$GasSwitch1 N001 N002 SW-TrPOS SW-TrGND IDSW
.ends switch_hcei
```



C.2 Brick

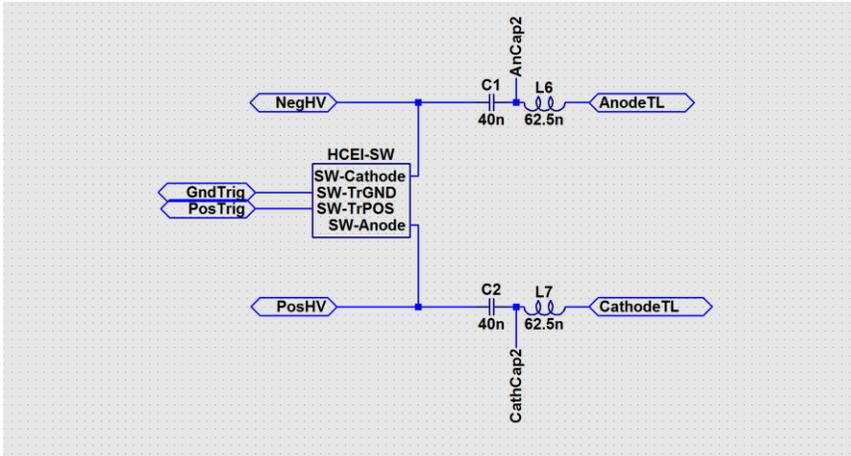
Netlist

```
.subckt ltd_brick_block AnodeTL CathodeTL GndTrig NegHV PosHV PosTrig
C1 AnCap2 NegHV 40n Rser=270m Lser=10n Rpar=0 Cpar=0
C2 CathCap2 PosHV 40n Rser=270m Lser=10n Rpar=0 Cpar=0
L6 AnodeTL AnCap2 62.5n Rser=0 Rpar=0 Cpar=0
```

```

L7 CathodeTL CathCap2 62.5n Rser=0 Rpar=0 Cpar=0
XHCEI-SW PosHV NegHV GndTrig PosTrig switch_hcei
.ends ltd_brick_block

```



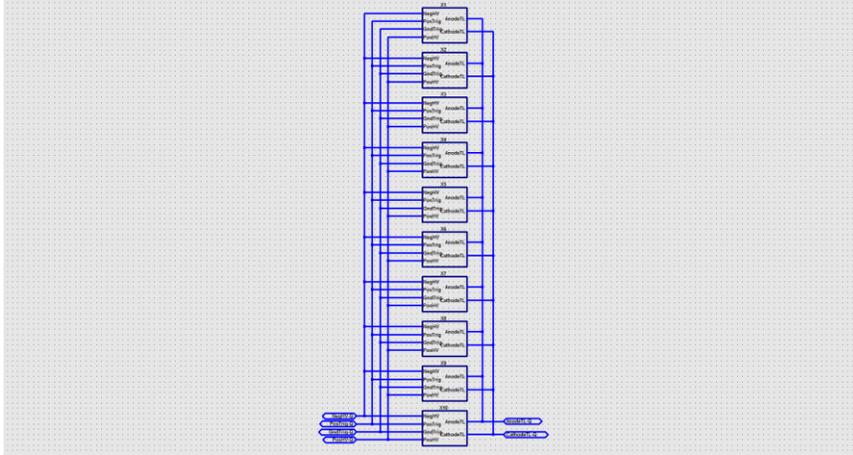
C.3 Quadrant

Netlist

```

.subckt ltd_quarter_block AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q
XX1 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX2 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX3 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX4 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX5 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX6 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX7 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX8 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX9 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX10 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
.ends ltd_quarter_block

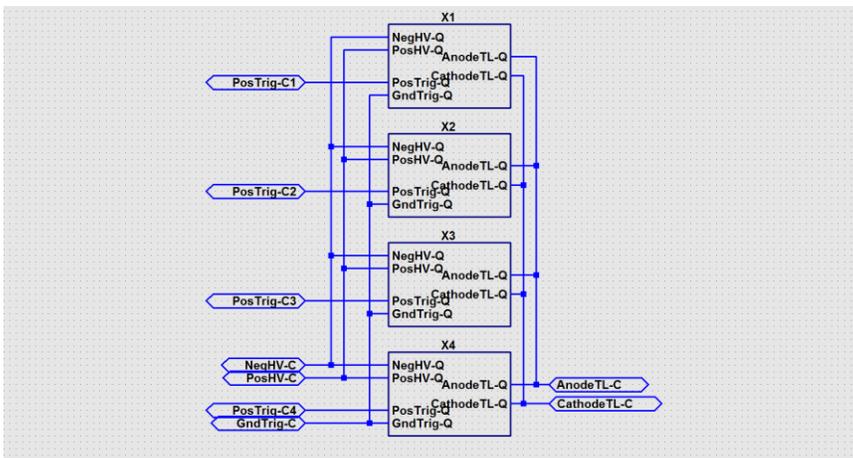
```



C.4 Module

Netlist

```
.subckt ltd_cavity_block AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C1 PosTrig-C2 PosTrig-C3 PosTrig-C4
XX1 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C1 ltd_quarter_block
XX2 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C2 ltd_quarter_block
XX3 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C3 ltd_quarter_block
XX4 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C4 ltd_quarter_block
.ends ltd_cavity_block
```



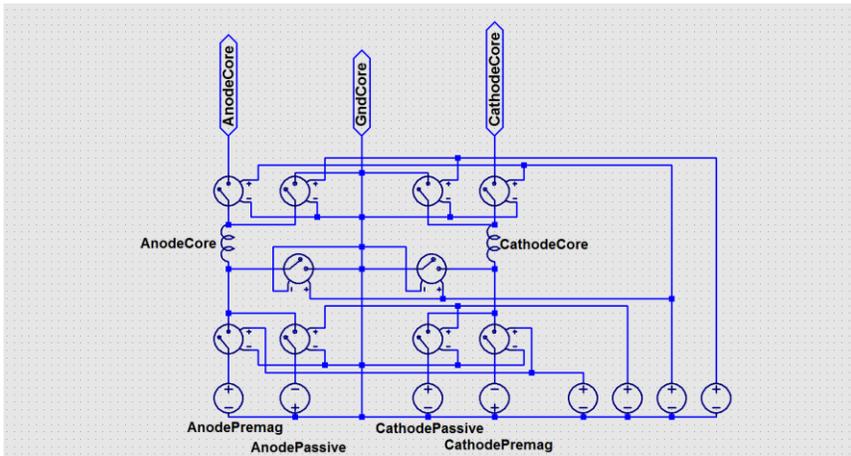
C.5 Premagnetizer

Netlist

```

.subckt ltd_premagpassive_block AnodeCore CathodeCore GndCore
L$AnodeCore N005 N003 Hc=30 Bs=2 Br=1.2 A=0.0032 Lm=6.28 Rser=0 Rpar=0.75 Lg=0 N=1
L$CathodeCore N006 N004 Hc=30 Bs=2 Br=1.2 A=0.0032 Lm=6.28 Rser=0 Rpar=0.75 Lg=0 N=1
V$CathodePremag GndCore N012 PULSE(0 12.8MEG 0 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay1 N012 N006 N008 GndCore IDSW
V$AnodePremag N009 GndCore PULSE(0 12.8MEG 0 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay2 N005 GndCore N002 GndCore IDSW
S$PosRelay3 N009 N005 N008 GndCore IDSW
S$PosRelay4 GndCore N006 N002 GndCore IDSW
V$PosPremag1 N008 GndCore PULSE(0 1 0 1p 1p 0.5n 0 1) Rser=1
V$PosPremag2 N002 GndCore PULSE(1 0 0 1p 1p 1n 0 1) Rser=1
S$PosRelay5 N004 CathodeCore N002 GndCore IDSW
S$PosRelay6 N003 AnodeCore N002 GndCore IDSW
V$AnodePassive GndCore N010 PULSE(0 5.12MEG 0.5n 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay7 N010 N005 N007 GndCore IDSW
V$CathodePassive N011 GndCore PULSE(0 5.12MEG 0.5n 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay8 N011 N006 N007 GndCore IDSW
V$PosPremag3 N007 GndCore PULSE(0 1 0.5n 1p 1p 0.5n 0 1) Rser=1
S$PosRelay9 N004 GndCore N001 GndCore IDSW
S$PosRelay10 N003 GndCore N001 GndCore IDSW
V$PosPremag4 N001 GndCore PULSE(0 1 0 1p 1p 1n 0 1) Rser=1
.ends ltd_premagpassive_block

```



C.6 Assembly

Netlist

```

XLTD-Cavity Anode CathStack 0 AnCap CathCap N002 N002 N002 N002 ltd_cavity_block
V$Trigger N002 0 PULSE(0 1 1n 1p 1p 2u)
V$NegHV 0 N004 70k
V$PosHV N003 0 70k
S$PosRelay N003 CathCap N001 0 IDSW
S$NegRelay N004 AnCap N001 0 IDSW

```

V\$PS-Connect N001 0 PULSE(1 0 1n 1p 1p 2u)
 L-load Anode ResLoad 4.5n Rser=0 Rpar=0 Cpar=0
 R-load ResLoad CathLoad 55m
 L\$Coaxial-TL N005 CathStack 8.65n
 L\$Radial-TL N006 N005 1f
 L\$RadFeed-TL N007 N006 1f
 L\$Planar-TL CathLoad N007 1f
 XCores Anode CathStack 0 ltd_premagpassive_block

* block symbol definitions

```
.subckt ltd_cavity_block AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C1 PosTrig-C2 PosTrig-C3 PosTrig-C4
XX1 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C1 ltd_quarter_block
XX2 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C2 ltd_quarter_block
XX3 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C3 ltd_quarter_block
XX4 AnodeTL-C CathodeTL-C GndTrig-C NegHV-C PosHV-C PosTrig-C4 ltd_quarter_block
.ends ltd_cavity_block
```

```
.subckt ltd_premagpassive_block AnodeCore CathodeCore GndCore
L$AnodeCore N005 N003 Hc=30 Bs=2 Br=1.2 A=0.0032 Lm=6.28 Rser=0 Rpar=0.75 Lg=0 N=1
L$CathodeCore N006 N004 Hc=30 Bs=2 Br=1.2 A=0.0032 Lm=6.28 Rser=0 Rpar=0.75 Lg=0 N=1
V$CathodePremag GndCore N012 PULSE(0 12.8MEG 0 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay1 N012 N006 N008 GndCore IDSW
V$AnodePremag N009 GndCore PULSE(0 12.8MEG 0 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay2 N005 GndCore N002 GndCore IDSW
S$PosRelay3 N009 N005 N008 GndCore IDSW
S$PosRelay4 GndCore N006 N002 GndCore IDSW
V$PosPremag1 N008 GndCore PULSE(0 1 0 1p 1p 0.5n 0 1) Rser=1
V$PosPremag2 N002 GndCore PULSE(1 0 0 1p 1p 1n 0 1) Rser=1
S$PosRelay5 N004 CathodeCore N002 GndCore IDSW
S$PosRelay6 N003 AnodeCore N002 GndCore IDSW
V$AnodePassive GndCore N010 PULSE(0 5.12MEG 0.5n 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay7 N010 N005 N007 GndCore IDSW
V$CathodePassive N011 GndCore PULSE(0 5.12MEG 0.5n 1p 1p 0.5n 0 1) Rser=0.75
S$PosRelay8 N011 N006 N007 GndCore IDSW
V$PosPremag3 N007 GndCore PULSE(0 1 0.5n 1p 1p 0.5n 0 1) Rser=1
S$PosRelay9 N004 GndCore N001 GndCore IDSW
S$PosRelay10 N003 GndCore N001 GndCore IDSW
V$PosPremag4 N001 GndCore PULSE(0 1 0 1p 1p 1n 0 1) Rser=1
.ends ltd_premagpassive_block
```

```
.subckt ltd_quarter_block AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q
XX1 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX2 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX3 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX4 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX5 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX6 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX7 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX8 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
```

```

XX9 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
XX10 AnodeTL-Q CathodeTL-Q GndTrig-Q NegHV-Q PosHV-Q PosTrig-Q ltd_brick_block
.ends ltd_quarter_block

```

```

.subckt ltd_brick_block AnodeTL CathodeTL GndTrig NegHV PosHV PosTrig

```

```

C1 AnCap2 NegHV 40n Rser=270m Lser=10n Rpar=0 Cpar=0

```

```

C2 CathCap2 PosHV 40n Rser=270m Lser=10n Rpar=0 Cpar=0

```

```

L6 AnodeTL AnCap2 62.5n Rser=0 Rpar=0 Cpar=0

```

```

L7 CathodeTL CathCap2 62.5n Rser=0 Rpar=0 Cpar=0

```

```

XHCEI-SW PosHV NegHV GndTrig PosTrig switch_hcei

```

```

.ends ltd_brick_block

```

```

.subckt switch_hcei SW-Anode SW-Cathode SW-TrGND SW-TrPOS

```

```

L1 SW-Anode N001 57.5n Rser=30m Rpar=0 Cpar=0

```

```

L2 N002 SW-Cathode 57.5n Rser=30m Rpar=0 Cpar=0

```

```

S$GasSwitch1 N001 N002 SW-TrPOS SW-TrGND IDSW

```

```

.ends switch_hcei

```

```

.tran 0 1001.002n 1.002n

```

```

.model IDSW SW(Ron=1e-9 Roff=1e9 Vt=.5)

```

```

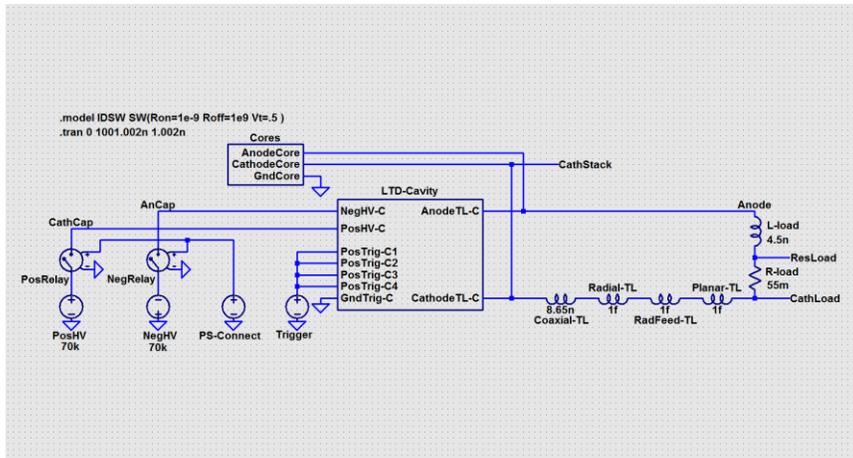
.backanno

```

```

.end

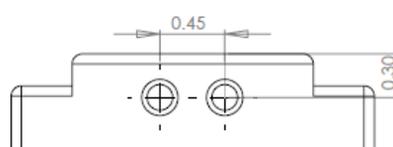
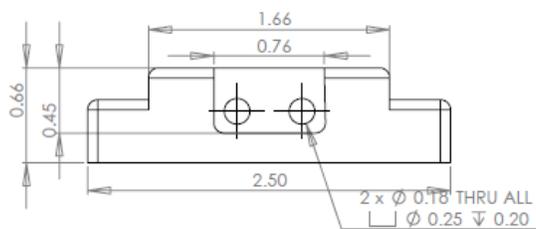
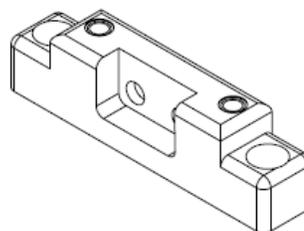
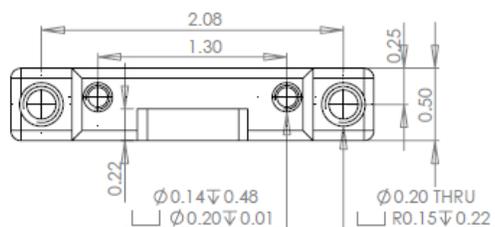
```



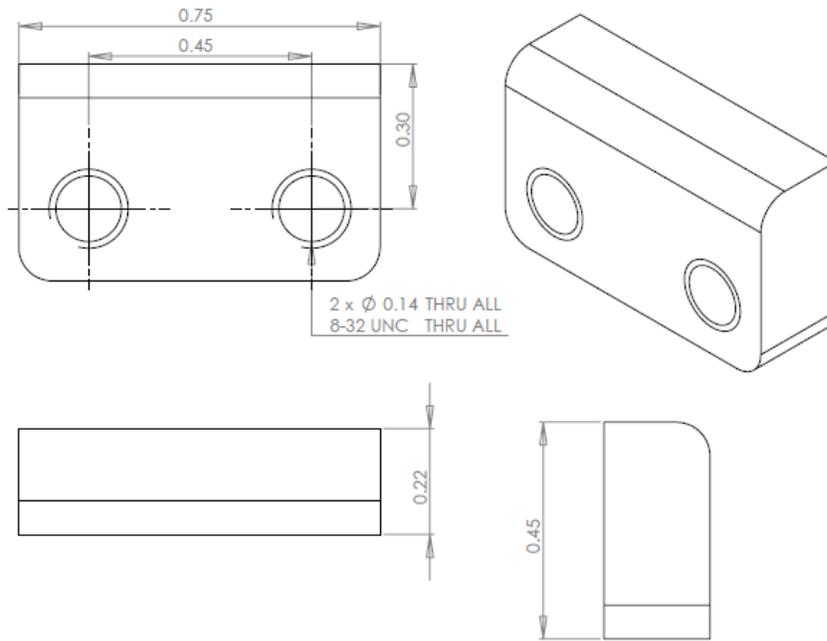
APPENDIX D

FOIL CARTRIDGE DESIGN

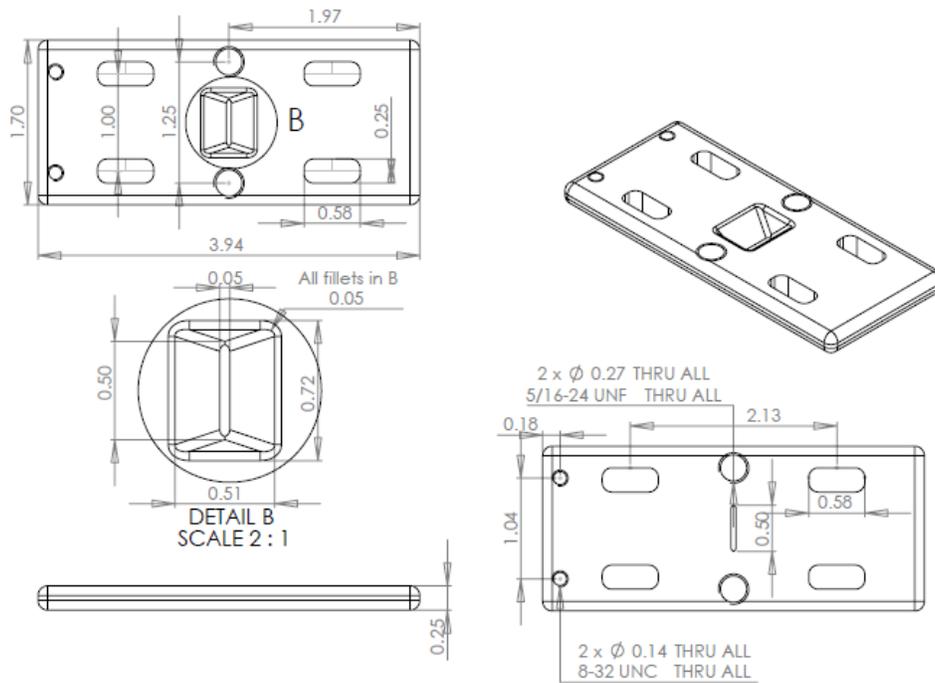
Cathode (Brass)
Fillets are all 0.07" unless noted.



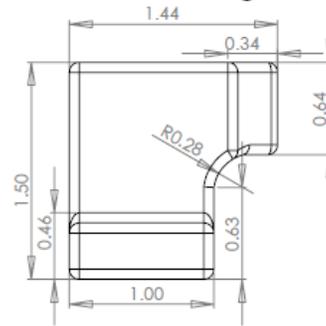
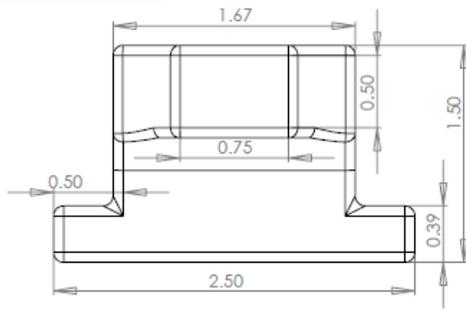
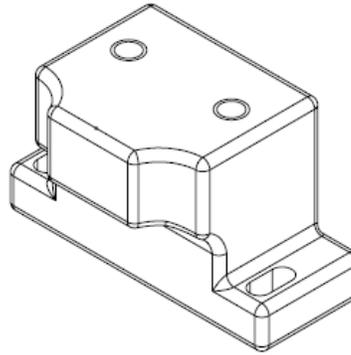
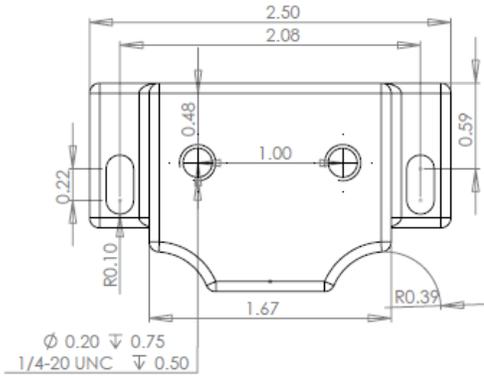
Cathode Inner Piece (Brass)
 Fillets are all 0.07" unless noted.



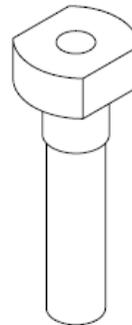
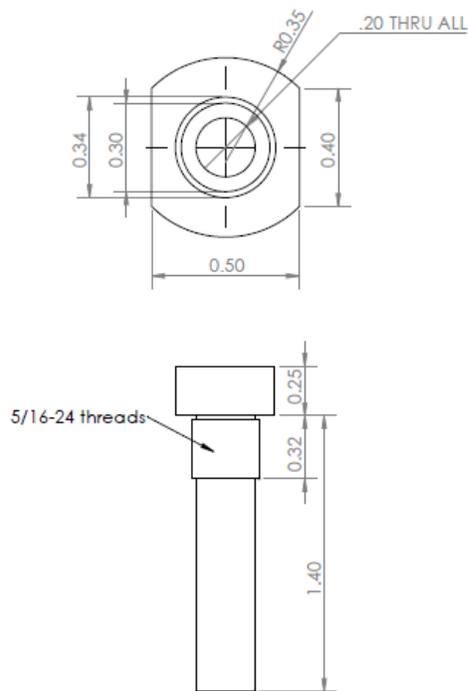
Central Plate (Brass)
 Fillets are all 0.07" unless noted.



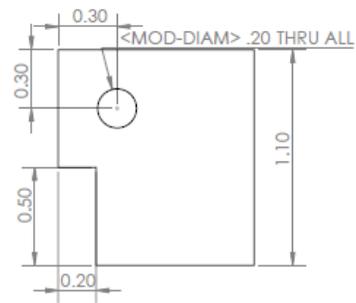
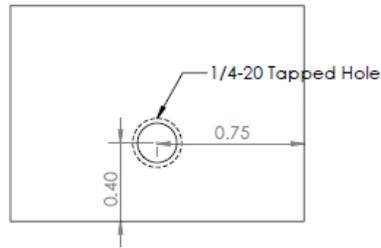
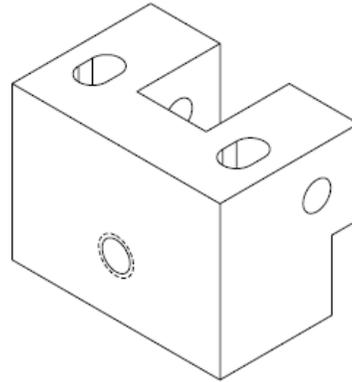
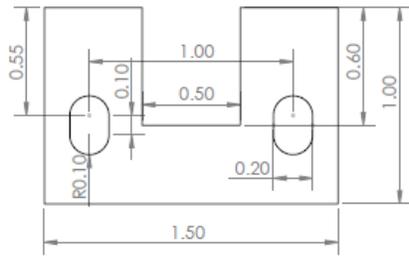
Anode x 2
 Fillets are all 0.07" unless noted.



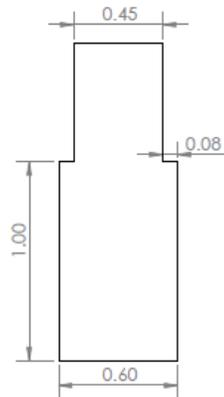
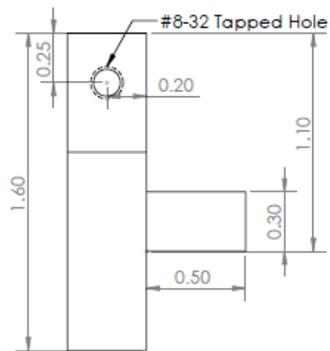
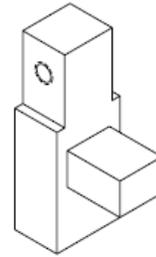
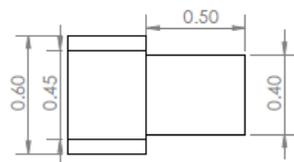
Posts x 2



Tension (plastic)



Tension (plastic)



APPENDIX E

EQUIPMENT INFORMATION

E.1 DAQ

DAQ Channel	Chip #	DAQ to Chip Pin #	Chip Pin # to Out	Supply Voltage (V)	Relay #	Panel Connector	Panel Pin #	Input/Output	Type	Usage
PFI 0 / P 1.0						BNC	1	I/O	Digital	Open
PFI 1 / P 1.1						BNC	2	I/O	Digital	Open
PFI 2 / P 1.2						BNC	3	I/O	Digital	Open
PFI 3 / P 1.3						BNC	4	I/O	Digital	Open
PFI 4 / P 1.4						BNC	5	I/O	Digital	Open
PFI 5 / P 1.5						BNC	6	I/O	Digital	Open
PFI 6 / P 1.6						BNC	7	I/O	Digital	Open
PFI 7 / P 1.7						BNC	8	I/O	Digital	Open
AI 0						BNC	9	Input	Analog	Negative LTD Voltage
AI 1						BNC	10	Input	Analog	Positive LTD Voltage
AI 2						BNC	11	Input	Analog	Laser Amplifier Voltage
AI 3						BNC	12	Input	Analog	Open
AI 4						BNC	13	Input	Analog	Open
AI 5						BNC	14	Input	Analog	Open
AI 6						BNC	15	Input	Analog	Open
AI 7						BNC	16	Input	Analog	Open
AI 8						BNC	17	Input	Analog	Open
AI 9						BNC	18	Input	Analog	Open
AI 10						BNC	19	Input	Analog	Open
AI 11						BNC	20	Input	Analog	Open
AI 12						BNC	21	Input	Analog	Open
AI 13						BNC	22	Input	Analog	Open
AI 14						BNC	23	Input	Analog	Open
AI 15						BNC	24	Input	Analog	Open
AO 0						BNC	25	Output	Analog	Open
AO 1						BNC	26	Output	Analog	Open
AO 2						BNC	27	Output	Analog	Open

AO 3						BNC	28	Output	Analog	Open
User 1						BNC	29	I/O	Digital	Open
User 2						BNC	30	I/O	Digital	Open
D GND						Case		GND	GND	GND
5 V +						NC	NC	Output	Power	NC
D GND						Case		GND	GND	GND
P 0.0	1	18	2	5		DB25	4	Input	Digital	Ground Monitor
P 0.1	1	16	4	5		DB25	5	Input	Digital	PS Connect Monitor +
P 0.2	1	14	6	5		DB25	6	Input	Digital	PS Connect Monitor -
P 0.3	1	12	8	5		DB25	7	Input	Digital	Area Interlock
D GND						Case		GND	GND	GND
P 0.4	1	9	11	5		DB25	8	Input	Digital	Equipment Interlock
P 0.5	1	7	13	5		DB25	9	Input	Digital	Ready
P 0.6	1	5	15	5		DB25	10	Input	Digital	Open
P 0.7	1	3	17	5		DB25	11	Input	Digital	Open
D GND						Case		GND	GND	GND
PFI 8 / P 2.0	2	18	2	5		Screw 1	1	Input	Digital	Open
PFI 9 / P 2.1	2	16	4	5		Screw 1	2	Input	Digital	Open
PFI 10 / P 2.2	2	14	6	5		Screw 1	3	Input	Digital	Open
PFI 11 / P 2.3	2	12	8	5		Screw 1	4	Input	Digital	Open
D GND						Case		GND	GND	GND
PFI 12 / P 2.4	2	11	9	5		Screw 2	1	Output	Digital	Open
PFI 13 / P 2.5	2	13	7	5		Screw 2	2	Output	Digital	Open
PFI 14 / P 2.6	2	15	5	5		Screw 2	3	Output	Digital	Open
PFI 15 / P 2.7	2	17	3	5		Screw 2	4	Output	Digital	Open
D GND						Case		GND	GND	GND
D GND						Case		GND	GND	GND
AI GND						Case		GND	GND	GND
AI Sense 1						Screw 1	9	Input	Analog	Open
AI Sense 2						Screw 1	10	Input	Analog	Open
CHS GND						Case		GND	GND	Chassis GND
P 0.8	3	18	2	5		DB25	12	Input	Digital	Open
P 0.9	3	16	4	5		DB25	16	Input	Digital	Open
P 0.10	3	14	6	5		DB25	17	Input	Digital	Open
P 0.11	3	12	8	5		DB25	18	Input	Digital	Open
D GND						Case		GND	GND	GND
P 0.12	4	2	18	8.3		DB25	19	Output	Digital	Open
P 0.13	4	4	16	8.3		DB25	20	Output	Digital	Open
P 0.14	4	6	14	8.3		DB25	21	Output	Digital	Open
P 0.15	4	8	12	8.3		DB25	22	Output	Digital	Open
D GND						Case		GND	GND	GND

P 0.16	4	11	9	8.3		DB25	23	Output	Digital	Power Supply Connect
P 0.17	4	13	7	8.3		DB25	24	Output	Digital	Purge
P 0.18	4	15	5	8.3		DB25	25	Output	Digital	Unabort
P 0.19	4	17	3	8.3	1	BNC	31	Output	Digital	Canon Camera Shutter
D GND						Case		GND	GND	GND
P 0.20	3	9	11	5		Screw 1	5	Input	Digital	Open
P 0.21	3	7	13	5		Screw 1	6	Input	Digital	Open
P 0.22	3	5	15	5		Screw 1	7	Input	Digital	Open
P 0.23	3	3	17	5		Screw 1	8	Input	Digital	Open
D GND						Case		GND	GND	GND
P 0.24	5	2	18	8.3		Screw 2	5	Output	Digital	Laser Amplifier Dump
P 0.25	5	4	16	8.3		Screw 2	6	Output	Digital	Laser Amplifier Charge
P 0.26	5	6	14	8.3		Screw 2	7	Output	Digital	Streak Camera Reset
P 0.27	5	8	12	8.3		Screw 2	8	Output	Digital	Open
D GND						Case		GND	GND	GND
P 0.28	5	11	9	8.3		Screw 2	9	Output	Digital	Open
P 0.29	5	13	7	8.3	2	BNC	32	Output	Digital	Streak Camera Reset
P 0.30	0	1		8.3		Chip 0		On/Off	Digital	Power On Low
P 0.31	0	2	18	8.3		Chip 0		On/Off	Digital	Power On High
CHS GND						Case		GND	GND	Chassis GND

E.2 Relay Control Unit

Pin #	Use	Pin #	Use
1	Area Interlock Panel In	25	PS Connect
2		26	
3	Equipment Interlock In	27	Unabort
4		28	
5	Ross Positive Relay Coil	29	Purge
6		30	
7	Ross Negative Relay Coil	31	Ready Signal
8		32	
9	Ross Ground Relay Coil	33	Area Interlock Signal
10		34	
11	Purge Valve	35	Equipment Interlock Signal
12		36	
13	+ Glassman Interlock	37	+HV Relay Signal
14		38	
15	- Glassman Interlock	39	-HV Relay Signal
16		40	
17	Charge Complete	41	Ground Relay

18		42	Signal
19	Ross 13 S1	43	
20	Ross 13 Hot	44	
21	Ross 13 S2	45	
22	Ross 14 S1	46	Ross 15 S1
23	Ross 14 Hot	47	Ross 15 Com
24	Ross 14 S2	48	Ross 15 S2

E.3 Voltage Monitor Calibrations

Monitor	Slope	Offset
+ Glassman	0.06978	-0.00085
- Glassman	0.06454	-0.00085
Laser Amplifier	-2.872	0

APPENDIX F

MRT EXPERIMENTAL PROCEDURES

F.1 Shot Checklist

PRE-SHOT

Vacuum chamber preparation

- _____ Lid closed with three bolts, visibly inspected to ensure good contact
- _____ Door flange under LTD closed
- _____ Roughing valve opened
- _____ TC Gauge plugged in, turned on
- _____ Roughing vacuum (< 80 mtorr) achieved
- _____ Roughing valve closed
- _____ Cryo valve opened
- _____ Ion gauge connected, powered on
- _____ Base pressure achieved and recorded, gauge readout unit removed
- _____ Foil tensioned

Laser preparation

- _____ Cooling water valve opened
- _____ Laser system and mode locker power on
- _____ Set heater to 5 and wait for oven light to turn off
- _____ Water cooling line cleared (water/ohms light no longer flashing)
- _____ Laser lamp powered on
- _____ Lamp ready
- _____ Mode locker set to 40.793 on auto (make sure oven temp is set first)
- _____ Q-switch powered on, set to external
- _____ Pockel's cell powered on
- _____ HV on Pockel's cell reached set level
- _____ Shutter opened
- _____ Laser diode output (cable 51) connected to oscilloscope, pulse train shape displayed

- _____ Neutral density removed from the beam path
- _____ Physical shutter opened
- _____ RF level set using oscilloscope and card
- _____ Q-switch trigger set to internal (at roughly 100 Hz); confirm Q-switched pulse train is ok
- _____ Cable 51 reconnected to bnc box
- _____ Laser beam path (up to physical beam block) verified using card
- _____ Physical shutter closed, announce “laser going into LTD room”
- _____ Physical beam block removed
- _____ Physical shutter opened

Laser alignment

- _____ Laser camera ISO set to 400, exposure time set
- _____ Laser verified on first mirror on optical table using card
- _____ Laser beamline verified on each mirror (moving beam block each time) up to targeting mirrors
- _____ Laser path established through LTD, visible on opposite side
- _____ Laser centered on camera, cutoff acceptably low
- _____ Camera mode set to bulb, ISO set to 100, external trigger cable connected

Spectrometer preparation

- _____ Gas cooling bottle opened
- _____ Water cooling valve opened
- _____ Cooling unit powered on
- _____ Position controller powered on
- _____ Cooling unit ready
- _____ PG 200 HV pulser powered on and proper settings established
- _____ Center wavelength set and verified
- _____ Winspec file path verified

Other preparations

- _____ _____ Neutral density installed in beam path before frequency doubler
- _____ Q-switch set to external
- _____ Flash lamp power supply powered on
- _____ Laser room door closed
- _____ Shutter opened on streak camera, make sure gate is set to internal and positive
- _____ Plug in switch pump
- _____ Room 112D checked to make sure charging circuit is functional; room locked upon exit
- _____ Turn on thyatron (at least 30 minutes prior to shot)

TEST SEQUENCE AND SHOT

- _____ Turn off all lights
- _____ Turn on interlock box
- _____ Verify all interlocks are set (gates closed, laser curtains in place, doors closed)
- _____ Turn on relay box
- _____ Turn on oscilloscopes and triggers
- _____ Turn on trigger circuitry for thyatron (below the spectrograph units)
- _____ Run trigger checks, verify all signals and delays
- _____ Turn on lights, check pre-shot image
- _____ Replace camera casing and close isolation canopy
- _____ Turn off LTD room lights
- _____ Purge switches
- _____ Raise switch and trigger pressures to the desired values
- _____ Turn on HV power supplies for test
- _____ Perform 5 kV test
- _____ Close door to screen room
- _____ After the desired thyatron warm-up time (usually 30 minutes), run LTD shot sequence

POST-SHOT

- _____ Turn off LTD HV supplies
- _____ Turn off laser HV supply
- _____ Purge switch/trigger lines
- _____ Close physical shutter on laser
- _____ Place beam blocks
- _____ Close cryo pump line
- _____ Open letup valve to let up LTD
- _____ Power off all spectrograph equipment, close water valve and note time
- _____ Close laser shutter
- _____ Turn off pulse picker and q-switch
- _____ Turn off mode locker
- _____ Turn off main laser power
- _____ Close water line
- _____ 30 minutes after cooling line for spectrograph is closed, close the spectrograph gas bottle
- _____ Unplug switch pump
- _____ Turn off thyatron

- _____ Turn off all cameras
- _____ Close shutter on streak camera
- _____ Transfer spectral and electrical data and images to file server
- _____ Premag to 3 kV
- _____ After LTD has let up, close the letup valve
- _____ Open the LTD and remove load hardware
- _____ Clean LTD surfaces and windows with kimwipes and q-tips
- _____ Sand target-facing fiber optic surfaces
- _____ Install the new foil in the load hardware and replace in the LTD
- _____ Close the lid; return to pre-shot checklist for the next shot

F.2 Settings

Trigger Device	Delay (μs)	Pulse Length (μs)	Amplitude (V)		Monitor Device	Delay (ns)
Thyratron	239.89	10	5		B-dot 2	88
Scopes	250.275	2	5		Switchyard PD	100
Q-switch	246.25	1	6		Laser Int. PD	161
Laser Amp.	0	1	5		Spectrograph	106
Spec Gate	250.86	1	5			

REFERENCES

- 1 K.S. Krane and D. Halliday, *Introductory Nuclear Physics* (Wiley, 1987).
- 2 J. Lindl, *Phys. Plasmas* **2**, 3933 (1995).
- 3 S.A. Slutz, M.C. Herrmann, R.A. Vesey, A.B. Sefkow, D.B. Sinars, D.C. Rovang, K.J. Peterson, and M.E. Cuneo, *Phys. Plasmas* **17**, 056303 (2010).
- 4 D.B. Sinars, S.A. Slutz, M.C. Herrmann, R.D. McBride, M.E. Cuneo, K.J. Peterson, R.A. Vesey, C. Nakhleh, B.E. Blue, K. Killebrew, D. Schroen, K. Tomlinson, A.D. Edens, M.R. Lopez, I.C. Smith, J. Shores, V. Bigman, G.R. Bennett, B.W. Atherton, M. Savage, W.A. Stygar, G.T. Leifeste, and J.L. Porter, *Phys. Rev. Lett.* **105**, 185001 (2010).
- 5 R.M. Gilgenbach, M.R. Gomez, J.C. Zier, W.W. Tang, D.M. French, Y.Y. Lau, M.G. Mazarakis, M.E. Cuneo, M.D. Johnston, B.V. Oliver, T.A. Mehlhorn, A.A. Kim, and V.A. Sinebryukhov, *AIP Conf. Proc.* **1088**, 259-262 (2009).
- 6 A.A. Kim, M.G. Mazarakis, V.A. Sinebryukhov, B.M. Kovalchuk, V.A. Visir, S.N. Volkov, F. Bayol, A.N. Bostrikov, V.G. Durakov, S.V. Frolov, V.M. Alexeenko, D.H. McDaniel, W.E. Fowler, K. LeChien, C. Olson, W.A. Stygar, K.W. Struve, J. Porter, and R.M. Gilgenbach, *Phys. Rev. ST Accel. Beams* **12**, 050402 (2009).
- 7 M.G. Mazarakis, W.E. Fowler, A.A. Kim, V.A. Sinebryukhov, S.T. Rogowski, R.A. Sharpe, D.H. McDaniel, C.L. Olson, J.L. Porter, K.W. Struve, W.A. Stygar, and J.R. Woodworth, *Phys. Rev. ST Accel. Beams* **12**, 050401 (2009).
- 8 M.G. Mazarakis, W.E. Fowler, K.L. LeChien, F.W. Long, M.K. Matzen, D.H. McDaniel, R.G. McKee, C.L. Olson, J.L. Porter, S.T. Rogowski, K.W. Struve, W.A. Stygar, J.R. Woodworth, A.A. Kim, V.A. Sinebryukhov, R.M. Gilgenbach, M.R. Gomez, D.M. French, Y.Y. Lau, J.C. Zier, D.M. VanDevalde, R.A. Sharpe, and K. Ward, *IEEE Trans. Plasma Sci.* **38**, 704-713 (2010).
- 9 I.H. Mitchell, J.M. Bayley, J.P. Chittenden, J.F. Worley, A.E. Dangor, M.G. Haines, and P. Choi, *Rev. Sci. Instrum.* **67**, 1533 (1996).
- 10 I. Smith, *Phys. Rev. ST Accel. Beams* **7**, (2004).
- 11 W.A. Stygar, M.E. Cuneo, D.I. Headley, H.C. Ives, R.J. Leeper, M.G. Mazarakis, C.L. Olson, J.L. Porter, T.C. Wagoner, and J.R. Woodworth, *Phys. Rev. ST Accel. Beams* **10**, (2007).
- 12 C. Deeney, C.A. Coverdale, M.R. Douglas, K.W. Struve, R.B. Spielman, W.A. Stygar, D.L. Peterson, N.F. Roderick, M.G. Haines, F.N. Beg, and J. Ruiz-Camacho, *Phys. Plasmas* **6**, 3576 (1999).
- 13 C. Deeney, M.R. Douglas, R.B. Spielman, T.J. Nash, D.L. Peterson, P. L'Eplattenier, G.A. Chandler, J.F. Seamen, and K.W. Struve, *Phys. Rev. Lett.* **81**, 4883 (1998).
- 14 M.K. Matzen, M.A. Sweeney, R.G. Adams, J.R. Asay, J.E. Bailey, G.R. Bennett, D.E. Bliss, D.D. Bloomquist, T.A. Brunner, R.B. Campbell, G.A. Chandler, C.A. Coverdale, M.E. Cuneo, J. Davis, C. Deeney, M.P. Desjarlais, G.L. Donovan, C.J. Garasi, T.A. Haill, C.A. Hall, D.L. Hanson, M.J. Hurst, B. Jones, M.D. Knudson, R.J. Leeper, R.W. Lemke, M.G. Mazarakis, D.H. McDaniel, T.A. Mehlhorn, T.J. Nash, C.L. Olson, J.L. Porter, P.K. Rambo, S.E. Rosenthal, G.A. Rochau, L.E. Ruggles, C.L. Ruiz, T.W.L. Sanford, J.F. Seamen, D.B. Sinars, S.A. Slutz, I.C. Smith, K.W. Struve, W.A. Stygar, R.A. Vesey, E.A. Weinbrecht, D.F. Wenger, and E.P. Yu, *Phys. Plasmas* **12**, 055503 (2005).
- 15 B. Jones, C.A. Coverdale, C. Deeney, D.B. Sinars, E.M. Waisman, M.E. Cuneo, D.J. Ampleford, P.D. LePell, K.R. Cochrane, J.W. Thornhill, J.P. Apruzese, A. Dasgupta, K.G. Whitney, R.W. Clark, and J.P. Chittenden, *Phys. Plasmas* **15**, 122703 (2008).
- 16 C.A. Coverdale, C. Deeney, P.D. LePell, B. Jones, J. Davis, R.W. Clark, J.P. Apruzese, J.W. Thornhill, and K.G. Whitney, *Phys. Plasmas* **15**, 023107 (2008).
- 17 M.E. Cuneo, R.A. Vesey, G.R. Bennett, D.B. Sinars, W.A. Stygar, E.M. Waisman, J.L. Porter, P.K. Rambo, I.C. Smith, S.V. Lebedev, J.P. Chittenden, D.E. Bliss, T.J. Nash, G.A. Chandler, B.B. Afeyan, E.P. Yu, R.B. Campbell, R.G. Adams, D.L. Hanson, T.A. Mehlhorn, and M.K. Matzen, *Plasma Phys. Control. Fusion* **48**, R1-R35 (2006).

- 18 J.H. Hammer, M. Tabak, S.C. Wilks, J.D. Lindl, D.S. Bailey, P.W. Rambo, A. Toor, G.B. Zimmerman, and J.L. Porter, *Phys. Plasmas* **6**, 2129 (1999).
- 19 G.A. Rochau, J.E. Bailey, Y. Maron, G.A. Chandler, G.S. Dunham, D.V. Fisher, V.I. Fisher, R.W. Lemke, J.J. MacFarlane, K.J. Peterson, D.G. Schroen, S.A. Slutz, and E. Stambulchik, *Phys. Rev. Lett.* **100**, (2008).
- 20 T.W.L. Sanford, G.O. Allshouse, B.M. Marder, T.J. Nash, R.C. Mock, R.B. Spielman, J.F. Seaman, J.S. McGurn, D. Jobe, T.L. Gilliland, M. Vargas, K.W. Struve, W.A. Stygar, M.R. Douglas, M.K. Matzen, J.H. Hammer, J.S. De Groot, J.L. Eddleman, D.L. Peterson, D. Mosher, K.G. Whitney, J.W. Thornhill, P.E. Pulsifer, J.P. Apruzese, and Y. Maron, *Phys. Rev. Lett.* **77**, 5063 (1996).
- 21 S.V. Lebedev, I.H. Mitchell, R. Aliaga-Rossel, S.N. Bland, J.P. Chittenden, A.E. Dangor, and M.G. Haines, *Phys. Rev. Lett.* **81**, 4152 (1998).
- 22 S.A. Pikuz, T.A. Shelkovenko, D.B. Sinars, J.B. Greenly, Y.S. Dimant, and D.A. Hammer, *Phys. Rev. Lett.* **83**, 4313 (1999).
- 23 T.A. Shelkovenko, S.A. Pikuz, J.D. Douglass, I.C. Blesener, J.B. Greenly, R.D. McBride, D.A. Hammer, and B.R. Kusse, *Phys. Plasmas* **14**, 102702 (2007).
- 24 D.H. Kalantar and D.A. Hammer, *Phys. Rev. Lett.* **71**, 3806 (1993).
- 25 J.D. Douglass, S.A. Pikuz, T.A. Shelkovenko, D.A. Hammer, S.N. Bland, S.C. Bott, and R.D. McBride, *Phys. Plasmas* **14**, 012704 (2007).
- 26 S.V. Lebedev, F.N. Beg, S.N. Bland, J.P. Chittenden, A.E. Dangor, M.G. Haines, K.H. Kwek, S.A. Pikuz, and T.A. Shelkovenko, *Phys. Plasmas* **8**, 3734 (2001).
- 27 J. Ruiz-Camacho, F.N. Beg, A.E. Dangor, M.G. Haines, E.L. Clark, and I. Ross, *Phys. Plasmas* **6**, 2579 (1999).
- 28 J.P. Chittenden, S.V. Lebedev, J. Ruiz-Camacho, F.N. Beg, S.N. Bland, C.A. Jennings, A.R. Bell, M.G. Haines, S.A. Pikuz, T.A. Shelkovenko, and D.A. Hammer, *Phys. Rev. E* **61**, 4370 (2000).
- 29 M. Haines, *Plasma Science, IEEE Transactions On* **30**, 588-592 (2002).
- 30 S.A. Pikuz, T.A. Shelkovenko, A.R. Mingaleev, D.A. Hammer, and H.P. Neves, *Phys. Plasmas* **6**, 4272 (1999).
- 31 J. Chittenden and C. Jennings, *Phys. Rev. Lett.* **101**, (2008).
- 32 S.V. Lebedev, F.N. Beg, S.N. Bland, J.P. Chittenden, A.E. Dangor, M.G. Haines, S.A. Pikuz, and T.A. Shelkovenko, *Phys. Rev. Lett.* **85**, 98 (2000).
- 33 C. Deeney, T.J. Nash, R.B. Spielman, J.F. Seaman, G.C. Chandler, K.W. Struve, J.L. Porter, W.A. Stygar, J.S. McGurn, D.O. Jobe, T.L. Gilliland, J.A. Torres, M.F. Vargas, L.E. Ruggles, S. Breeze, R.C. Mock, M.R. Douglas, D.L. Fehl, D.H. McDaniel, M.K. Matzen, D.L. Peterson, W. Matuska, N.F. Roderick, and J.J. MacFarlane, *Phys. Rev. E* **56**, 5945 (1997).
- 34 C. Coverdale, C. Deeney, M. Douglas, J. Apruzese, K. Whitney, J. Thornhill, and J. Davis, *Phys. Rev. Lett.* **88**, (2002).
- 35 M.G. Mazarakis, C.E. Deeney, M.R. Douglas, W.A. Stygar, D.B. Sinars, M.E. Cuneo, J. Chittenden, G.A. Chandler, T.J. Nash, K.W. Struve, and D.H. McDaniel, *Plasma Devices & Operations* **13**, 157-161 (2005).
- 36 S.N. Bland, S.V. Lebedev, J.P. Chittenden, G.N. Hall, F. Suzuki-Vidal, D.J. Ampleford, S.C. Bott, J.B.A. Palmer, S.A. Pikuz, and T.A. Shelkovenko, *Phys. Plasmas* **14**, 056315 (2007).
- 37 D. Sinars, M. Cuneo, E. Yu, D. Bliss, T. Nash, J. Porter, C. Deeney, M. Mazarakis, G. Sarkisov, and D. Wenger, *Phys. Rev. Lett.* **93**, (2004).
- 38 D.B. Sinars, M.E. Cuneo, B. Jones, C.A. Coverdale, T.J. Nash, M.G. Mazarakis, J.L. Porter, C. Deeney, D.F. Wenger, R.G. Adams, E.P. Yu, D.E. Bliss, and G.S. Sarkisov, *Phys. Plasmas* **12**, 056303 (2005).
- 39 E.P. Yu, M.E. Cuneo, M.P. Desjarlais, R.W. Lemke, D.B. Sinars, T.A. Haill, E.M. Waisman, G.R. Bennett, C.A. Jennings, T.A. Mehlhorn, T.A. Brunner, H.L. Hanshaw, J.L. Porter, W.A. Stygar, and L.I. Rudakov, *Phys. Plasmas* **15**, 056301 (2008).
- 40 B. Jones, C.J. Garasi, D.J. Ampleford, C. Deeney, T.A. Mehlhorn, S.N. Bland, S.V. Lebedev, J.P. Chittenden, S.C. Bott, J.B.A. Palmer, G.N. Hall, and J. Rapley, *Phys. Plasmas* **13**, 056313 (2006).
- 41 G.N. Hall, S.N. Bland, S.V. Lebedev, J.P. Chittenden, J.B.A. Palmer, F.A. Suzuki-Vidal, G.F. Swadling, N. Niasse, P.F. Knapp, I.C. Blesener, R.D. McBride, D.A. Chalenski, K.S. Bell, J.B. Greenly, T. Blanchard, H. Wilhelm, D.A. Hammer, B.R. Kusse, and S.C. Bott, *AIP Conf. Proc.* **1088**, 89-94 (2009).
- 42 J.P. Chittenden, S.V. Lebedev, B.V. Oliver, E.P. Yu, and M.E. Cuneo, *Phys. Plasmas* **11**, 1118 (2004).

- 43 J.B. Greenly, J.D. Douglas, D.A. Hammer, B.R. Kusse, S.C. Glidden, and H.D. Sanders, *Rev. Sci. Instrum.* **79**, 073501 (2008).
- 44 J.C. Zier, J.D. Douglass, I.C. Blesener, K.S. Blesener, D.A. Chalenski, R.M. Gilgenbach, J.B. Greenly, D.A. Hammer, P.F. Knapp, B.R. Kusse, Y.Y. Lau, R.D. McBride, W. Syed, and E.P. Yu, *Phys. Plasmas* **16**, 102702 (2009).
- 45 T.A. Shelkovenko, D.B. Sinars, S.A. Pikuz, K.M. Chandler, and D.A. Hammer, *Rev. Sci. Instrum.* **72**, 667 (2001).
- 46 J.D. Douglass and D.A. Hammer, *Rev. Sci. Instrum.* **79**, 033503 (2008).
- 47 I.M. Rittersdorf, Y.Y. Lau, J.C. Zier, R.M. Gilgenbach, E.J. Cruz, and J.W. Luginsland, *Appl. Phys. Lett.* **97**, 171502 (2010).
- 48 F.N. Beg, J. Ruiz-Camacho, M.G. Haines, and A.E. Dangor, *Plasma Phys. Control. Fusion* **46**, 1-10 (2004).
- 49 G. Taylor, *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* **201**, 192-196 (1950).
- 50 E.G. Harris, *Phys. Fluids* **5**, 1057 (1962).
- 51 S. Chandrasekhar, *Hydrodynamic and Hydromagnetic Stability* (Courier Dover Publications, 1981).
- 52 P.M. Bellan, *Fundamentals of Plasma Physics* (Cambridge University Press, 2006).
- 53 M. Mazarakis, W. Fowler, K. LeChien, F. Long, M. Matzen, D. McDaniel, R. McKee, C. Olson, J. Porter, S. Rogowski, K. Struve, W. Stygar, J. Woodworth, A. Kim, V. Sinebryukhov, R. Gilgenbach, M. Gomez, D. French, Y. Lau, J. Zier, D. VanDevalde, R. Sharpe, and K. Ward, *Plasma Science, IEEE Transactions On* **38**, 704-713 (2010).
- 54 M.R. Gomez, R.M. Gilgenbach, Y.Y. Lau, W. Tang, J.C. Zier, M.G. Mazarakis, M.E. Cuneo, T.A. Mehlhorn, and W.A. Stygar, in *Pulsed Power Conference, 2007 16th IEEE International* (2007), pp. 152-155.
- 55 A.A. Kim, *Development, Production and Test of a Fast LTD Stage Type Z* (Institute of High Current Electronics, 2006), pp. 1-46.
- 56 M. Engelhardt, *Ltspice* (Linear Technology Corporation, 2010).
- 57 J. Ball and A. Thurtell, *LTD Controller User's Manual* (University of Michigan, 2010).
- 58 E. Hecht, *Optics* (Addison-Wesley, 2002).
- 59 R.B. Spielman, C. Deeney, G.A. Chandler, M.R. Douglas, D.L. Fehl, M.K. Matzen, D.H. McDaniel, T.J. Nash, J.L. Porter, T.W.L. Sanford, J.F. Seamen, W.A. Stygar, K.W. Struve, S.P. Breeze, J.S. McGurn, J.A. Torres, D.M. Zagar, T.L. Gilliland, D.O. Jobe, J.L. McKenney, R.C. Mock, M. Vargas, T. Wagoner, and D.L. Peterson, *Phys. Plasmas* **5**, 2105 (1998).
- 60 F.F. Chen, *Introduction to Plasma Physics and Controlled Fusion: Plasma Physics* (Springer, 1984).
- 61 A. Gerrard and J.M. Burch, *Introduction to Matrix Methods in Optics* (Courier Dover Publications, 1994).
- 62 G.S. Settles, *Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media* (Springer, 2001).