Draft of HIT-SI3 Ion Doppler Spectrometer Manual

Rian Chandra[1] and Aaron Hossack, Ph.D[1]

[1]University of Washington

April 28, 2017

Quick Overview

- Confirm that the IDS fiber or fibers are inserted, locked, and oriented in the configuration required, and that the spectrometer is dialed to the wavelength of interest (usually λ=464.97nm, for the CIII doublet).
- Check to make sure that the camera is turned on (lights on back are on), and that the camera computer can communicate with it (open either Phantom Camera Control program on the IDS camera operator computer, or the older one (Ph692, shortcut on desktop), which is somewhat more intuitive). Check to make sure that the Framerate, Resolution, and Exposure are correct, in that order.
- Open Matlab on the IDS operator computer. Add the entire PhMatlabSDK file and its contents to the search path, and then remove `T:\RChandra\Phantom\PhMatlabSDK\bin\win32`. This file must be explicitly removed from the path.
- Run `C:\Users\HITSI\Documents\GitHub\IDS_Rian\PhantomStalker.m`. In theory, it should automatically pull the shot number from the tree, convert the cine from the camera, and save it, and the converted file. Sometimes you need to restart it if it has a hard time. It won't work if a baseline shot doesn't exist. Exposure and other parameters can be set in Phantom Camera Control as needed.
- The .mat file will be saved in the Imaging3 tree, and in `T:\IDS\Data_Repository\Shot[shot number].mat`. PDC shots will not save to the Imaging3 tree.

Contents

1 Hardware overview and calibration

1.1 General notes:

- Wearing protective glasses is necessary when using the mercury lamp (these glasses have blue frames). A sign should be put on the door if the lamp is in use to warn others.

- The fibers are not aligned perfectly vertically with the entrance slit. We expect that this will manifest as a second degree polynomial dependence (wavelength vs real space) on the CCD (linear fit $\times$ optics curvature). See Cal_Main 2b, and Sec.3.1.

- The motor speed is not perfectly even. See Cal_Main 5c ans Sec.3.1 for correcting this with a known doublet. If the motor is sufficiently slow (or the camera frame rate sufficiently high), the stepper-nature becomes apparent. The speed is also slightly periodic.

- The tighter the entrance slit, the lower the instrument temperature, the lower the signal.

1.2 Image Intensifier

The image intensifier has not been used to date on the HIT-SI3 experiment. It was used for some high frequency data collection on HIT-SI, and currently resides in the lower shelf of the control room optics cabnet. It has an unfortunately low duty cycle, with an on time of 3-7μs and an off time of $\approx 100\mu$s. See old IDS manual for more details.

1.3 About the spectrometer

IDS currently uses a 1m focal length, Czerny-Turner configuration spectrometer, with a wavelength range of approximately 250-700 nm, and a focal length of f/8.5, on long term loan from professor Nagata at the university of Hyogo. The spectrometer includes a motor for automatic adjustment of the grating, which is useful for calibration. The camera lens, and entrance and exit slits are adjustable. The cart should be docked with pegs down to

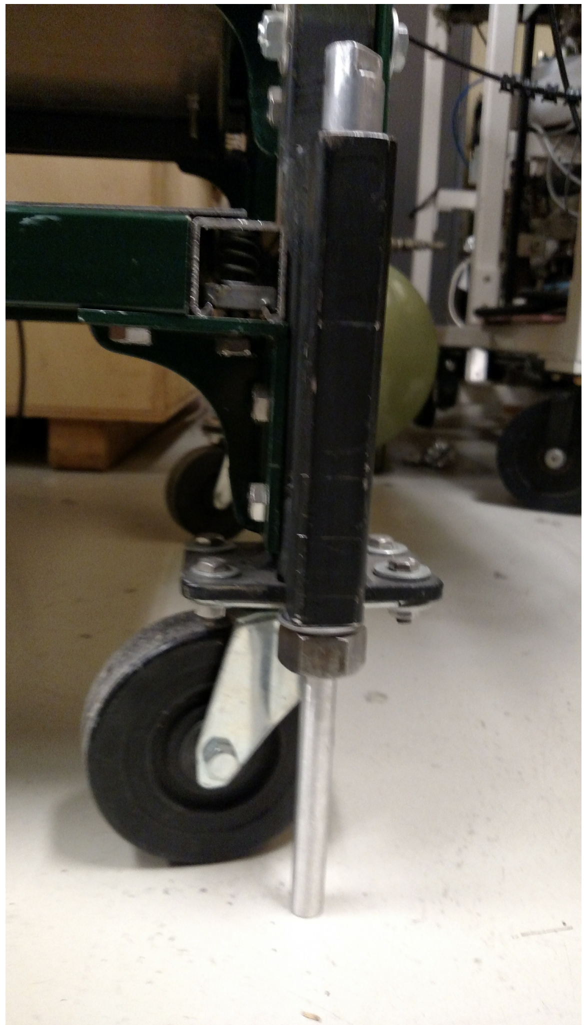

Figure 1: IDS Cart Breaks Engaged

stabilize, as in Fig. 1.

The two 3m fiber optic cables attached to the entrance slit (Fig. 2) both contain linear arrays of 36 individual fibers each. A fisheye lens at the end of each fiber separates each light-cone by approximately 2.9^o.If placed in a midplane port, the diameter of the light cone going through the geometric axis will be $\approx$ 5.1cm at the wall. The aluminum fiber holders are machined to fit into the re-enterant ports.

The light from each fiber is optically coupled to the CCD of a Phantom v710 fast camera from Vision Research International. A typical CCD frame is given in Fig 4 along with the spectral lines of typical interest. The camera can be plugged into the "Cart Computer" (Fig 3) for calibration purposes. This computer will not start unless an active

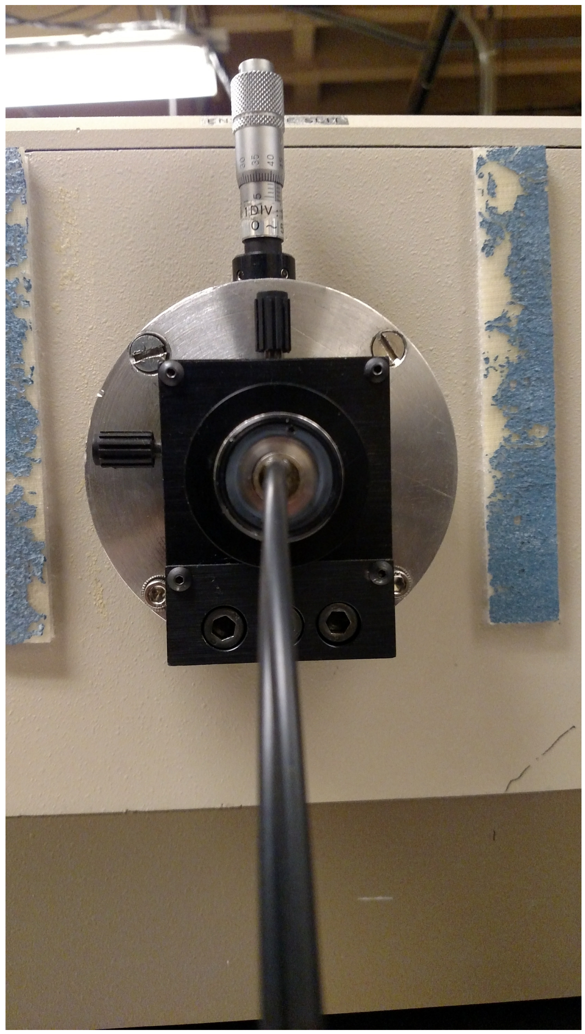

Figure 2: Spectrometer Entrance Slit

Figure 3: Lab Cart Computer, Useful for Calibration

Ethernet cord is plugged into the network card on the back. No one knows why this is.

1.4 Calibrating PIX_SP

What: PIX_SP gives [nm]/[pixel] for each fiber channel. When properly calculated, it's average value is $\approx 1.1 \times 10^{-11}$ [m]/[pixel].

Why: because we need to convert size and shift of the light gaussian on the CCD to wavelengths.

How: Setup one fiber on the test stand as pictured in Fig 5. Setup motor box (plug in connections). Set motor speed to $\leq$.1nm/s. Dial spectrometer to 435.6nm (the brightest Mg line). With the mobile camera cart connected to the camera through the network card,

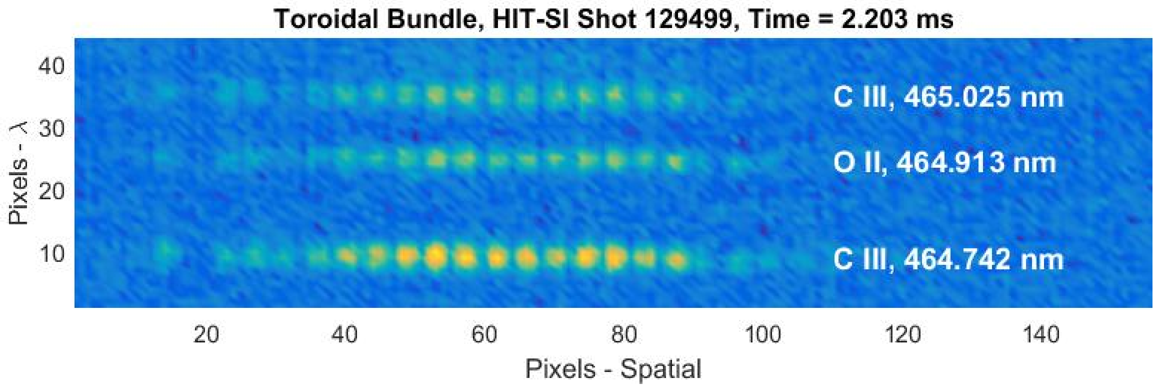

Figure 4: Typical Spectral Lines on the CCD, HITSI Shot 129499

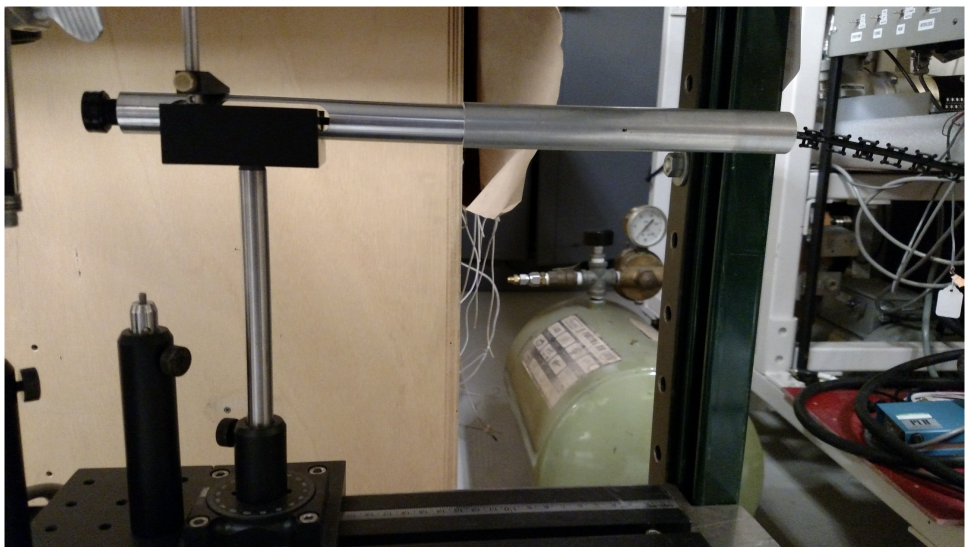

Figure 5: Fiber in the holder for calibration

open the Ph692. Set the camera to continuous recording, trigger it, and immediately turn on the motor, running down (lambda decreases as time goes forward). You should see the main line, and the a few seconds later a secondary, weaker line (and potentially a third line). Stop the movie after the second line has past. See Sec.5.1 for more details on running the recording software.

1.5 Calibrate PEAKS

What: PEAKS gives the location on the CCD of the center of each light-gaussian, along with its width ($\sigma_{x,y}$).

Why: Because we need to know where the centers and zero temperature widths are, relatively, between all the fibers, to compare with during a shot. This is particularly important if using both fibers in the midplane.

How: Set a fiber in the correct orientation in the test stand. Dial spectrometer to 435.6nm, set Ph692 to continuous recording, trigger, and record for several seconds. Save the file. Do this for each fiber.

1.6 Backlighting the Fibers

Why: To confirm that the fiber lens is correctly focused.

How: Use the arts and crafts box (Fig. 6, which lives above the cabinet which holds old PMTS and is next to the optics cabinet. The string is approximately the length of the major radius. Set up the fiber in the holder. Place the box directly in front of the fiber, one string length away. Remove the spectrometer top and middle panels from the entrance slit. Place the 1ft long, 2in diameter black laser (sometimes used for Thompson calibration) inside the middle panel, raised by 1-2 in. Adjust as necessary so that circles of light appear inside the box, adjust the lens on the fiber to focus these circles. If necessary, rapidly swinging the laser up and down can be used to align the fiber holder in an absolute vertical position.
Note: Critical care should be taken whenever the spectrometer panels are removed, and when working inside it.

1.7 Finding Chord Angles

Why: This information goes into calc_geometry_5 to find the impact parameter of each chord.

How: Setup the fiber holder on the stand, opposite the mercury lamp, and zero the angle ticks. Find the maximum and minimum angle that illuminates the first and last fibers. Enter this range into calc_geometry_5.

Figure 6: The "Arts and Crafts" Box Resides Above the Optics Cabinet in the Control Room

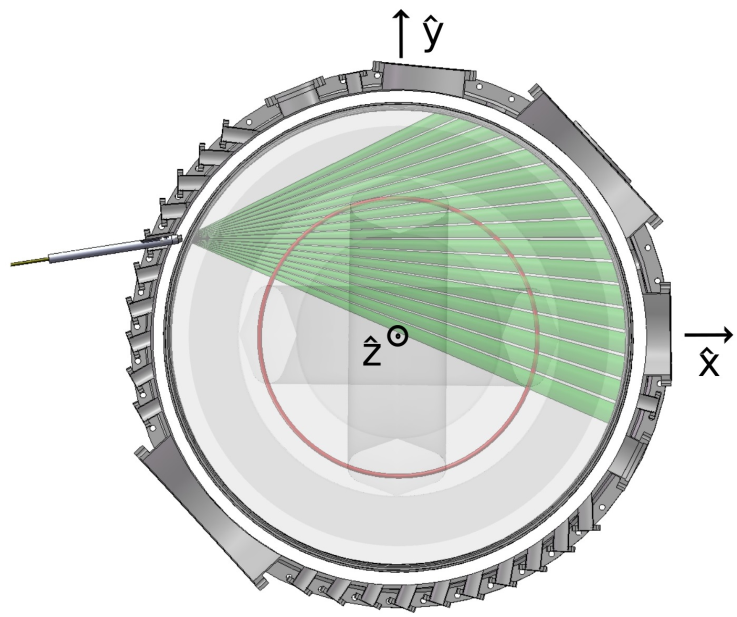

Figure 7: Upper Mohawk 6 Port, used on HIT-SI

2 Hardware Setup During Operations

2.1 Fiber Orientation

Directly after calibration is finished, the fiber holders should be carefully returned to the re-entrant ports, maintaining vertical alignment. Markings on the sides of the holders, and the port entrances should indicate proper alignment. The holders should be inserted with a small amount of space at the very end (ie: not pressed directly up against the glass). If required, the impact parameters for the fibers in the midplane, rotated 90^o have been calculated.

2.2 Port Options and Names

- Upper mohawk 6: Views upper half of midplane, views tangentially though impact parameters, Fig 7.
- Lower mohawk 6: Inverse of upper mohawk 6, rotated below it by $\approx 90^o$. Fig 8.
- 71 degree radial port: Used for some data in Aaron's thesis Fig 9.

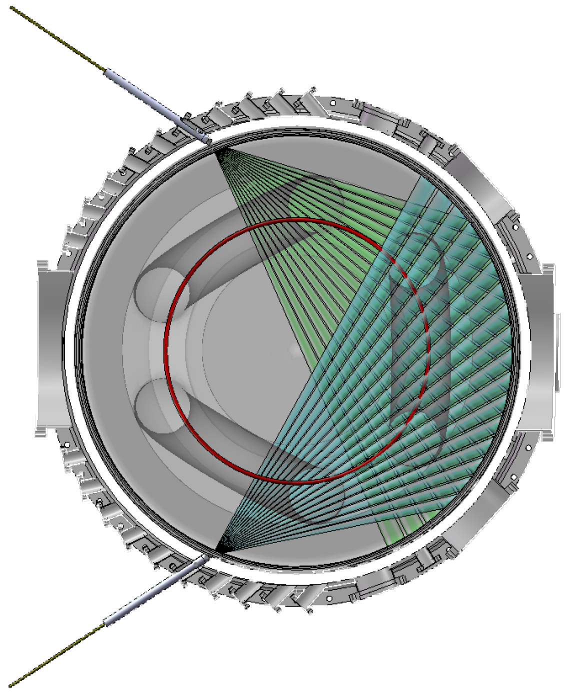

Figure 8: Both Mohawk Ports ± 6, used on HIT-SI3

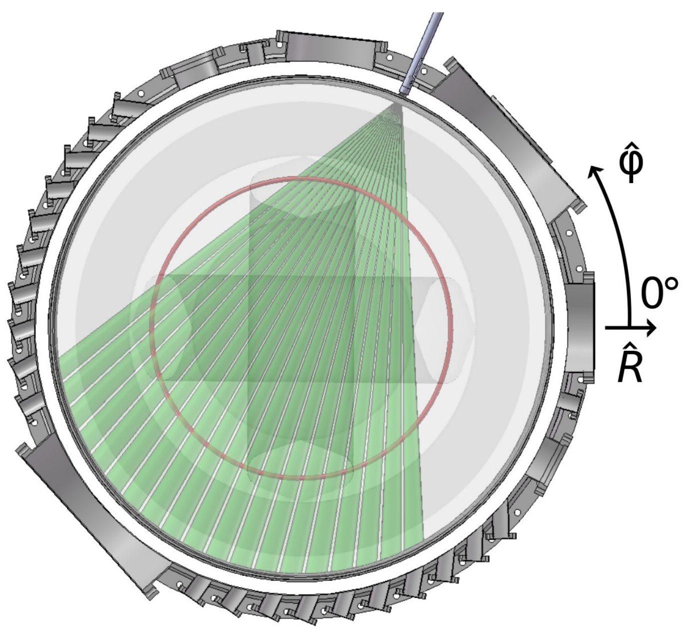

Figure 9: Radial 71° Port

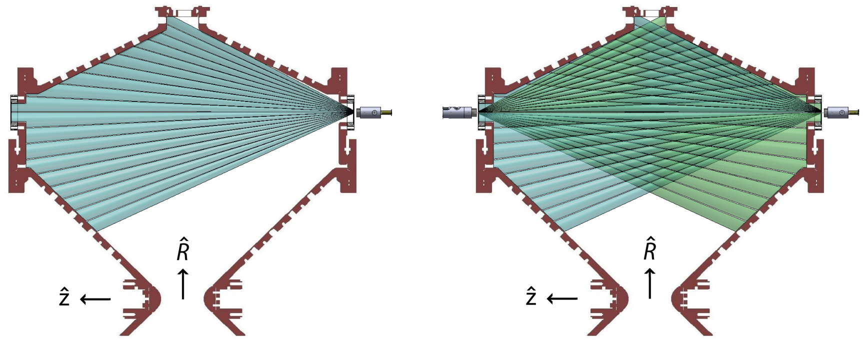

Figure 10: Dual-Facing Poloidal Ports, for Absolutely Specifying Poloidal Velocity

- Brian Victor Coherent fiber ports: Useful for taking movies of the plasma with the coherant fiber array. Not usually used for IDS data.
- Poloidal ports: Can be used to view the same poloidal cross section with two fibers facing eachother, Fig 10.

3 Calibration software steps

3.1 What each phase of calibration does

The calibrationMAIN code is relatively well documented. It takes in the calibration movies and some additional information and stores the reference values the tree for a specified region of shots, or the model tree shot (-1). Phases 1-2 take in the stationary fiber movie(s), apply biorthogonal decomposition to filter them, and calculate the width and x,y locations of each gaussian based on automatically calculated initial guesses. Phases 3-4 are depreciated. Phases 5a,c,d perform an initial calculation of the wavelength-per-pixel (PIX_SP), and then corrects it by explicitly finding the motor speed. Phase 5d corrects for the spectrometer dial moving sligtly as fiber bundles are changed out during the calibration, which can lead to a discontinuity in the graph of y_0 vs channel. Phase 5b is depreciated. Phase 6 calculates the relative intensity of each channel, and plots the calibration values. Phase 7 removes dead channels. Phase 8 saves the requested calibration parameters to the tree. The values collectively saved by calibrationMAIN are organized by fiber channel.

PEAKS column one links channel number to row index.

3.2 What the calibration saves to the tree

- CAL_LAMBDA: this should be the wavelength, during data acquisition, that the calibrated line will be set to. When doing this, the assumption that the spectrometer grating behaves roughly linearly is made. (IE: that PIX_SP is the same, even at a slightly different wavelength). This is used to calculate the expected instrument temperature for the lines being viewed.

- LAMBDA: The wavelengths, in meters, of the lines being viewed for data acquisition. This is necessary for predicting the relative locations of the observed lines on the CCD. It is also necessary when calculating temperature.

- VOLTAGE: The voltage of the Image Intensifier. Not technically depreciated, but rarely used.

- MASS: Ion mass of the viewed lines, in AMU. Necessary for calculating velocity and temperature

- PEAKS: In the following column order: Channel number, the x center (necessary for obvious reasons) and y center (delta y gives us velocity)., the x width of the gaussian (which is currently only used to improve the initial guess for data fitting), and the y width (the change in which gives us temperature).

- IMPACTS: Generated by calc_geometry5, impacts gives the impact parameter corresponding to each channel. This is specific to whichever port and orientation the fibers are in, and so may change between calibrations.

4 The Tree

4.1 How the tree works:

Useful References: PhantomStalker, SaveToTree3, LoadParams, BatchCorrect2, The MDSplus online tutorial.

The HITSI3 tree, and the ANALYSIS3 subtree, and IMAGING3 subtree store the experimental and calibration data for IDS, and now the raw data from the movies as well. Data is stored in formats specific to MDSplus. The tree structure can be visually navigated via the "Traverser" program, installed on most lab computers. This is a useful way to get a feel for the layout. The JScope program is used to display signals from the tree. This is not terribly useful for IDS at present. In theory, JScope can read the raw movies from the camera, stored as MDSplus "segments", but this has not been tested as yet.

Each shot has its unique hitsi3, analysis3, and imaging3 tree (the latter two can be accessed from the hitsi3 tree). Data is stored in "Nodes", and is usually "Numeric" type. Nodes can either be referenced from the top of the tree, or through a tag, such as '`\IDS_MASS`'. Any data entry, or node traversing will be referenced from the current node.

4.2 How to access data in the tree:

There are three different methods of accessing data from the tree, all of which are slightly nuanced. The "connection object" (thin client connection) is the newest way to connect to the tree. It has been found to be the most straightforward way to do basic data entry and access. Modification of the tree structure itself is best accomplished(/only possible) using the "Tree Object" (Object Oriented MDSplus). Performing mathematical operations, such as "`sihi_smooth`", or more complex TDI expressions, can be accomplished in the Thin Client, but is easiest in the mdsconnect (TDI expression) framework.
Examples of these three methods can be found in the following codes:

- Thin Client: loadParams.m, saveToTree3.m, both in `\IDS\NewCodes` have good examples of this
- Object Oriented PhantomStalker.m in `\RChandra`, saveToTree3.m in `\IDS\Calibration`
- mdsconnect Not currently used in any up-to-date IDS code, exists in some depreciated codes.

4.3 Tree Notes:

- Note the difference between opening a tree, and opening it for ‘EDIT’. This should only be done when absolutely necessary, and can only be done on the “hitsi” network account.

- Environment variables and path: Check a computer which can successfully access the tree, and make sure that the environment variables on the machine you are using include analysis3_path, hitsi3_path, and imaging3_path. Remove the 3 if looking at hitsi data. Matlab also needs to know where the MDSplus libraries are stored. Note that import in matlab will unfortunately not throw an error if the imported library is not found. The MDSplus tutorial has the correct path locations. This may require modifying the javaclasspath.txt file, or the librarypath.txt. It may be easier to manually add the paths in the matlab GUI, and then use savepath to permanently add the files to your path on startup.

- Directly on import (either from the Tree object or the Connection object), the data will be of type MDS_(float32/uint/float32array/etc). If all of the MDSplus Matlab files are on the path, `NATIVEvalue` from MDSplus or `double` from Matlab will convert to a usual type.

- When adding data into the tree, clear the node first as a precaution (see SaveToTree3). When adding arrays, you have to add the data as `MDSargs(data)`, or you will be instead adding `squeeze(data)`, and everything will be a vector.

- See PhantomStalker and SaveToTree3 for examples of how to move around in the tree.

- Occasionally, when using the Tree Object connection, if too many calls to the tree are made in a row (N$\gtrsim$50), MDSplus will throw an error, and you will need to restart Matlab. The Connection Object does not appear to have this issue.

5 Data Collection Software:

5.1 How to run the PCC

Assuming the camera is connected via the fiberoptic ethernet adaptor, it can be controlled from the IDS camera operator computer by way of the Phantom Camera Control software, accessible from the desktop. In the Ph692 program, $->$ Acquisition $->$ Setup and Recording opens a screen where the current camera image can be viewed in real time, and acquisition parameters can be set. To prepare the camera for data acquisition, click Continuous Recording, and the camera will wait for the server to send out the trigger signal.

Important Notes on the PCC:

- The framerate should be changed to be some integer multiple of the injector frequency. The exposure should be maximized given this frame rate. Do not attempt to optimize based first on exposure. This will lead to beat frequencies between the frame rate and the injector period.

- After it has been triggered, closing the acquisition window will open a viewer for the currently recorded cine. It is absolutely critical that if any image processing parameters are changed, they are reset to the default before the cine is converted into a .mat file. If not, these changes will be propagated through, and the .mat file will not be useable. Explanations of how the image processing parameters affect the movie can be found in the Cine Documentation.

- If there are too many PostTrigger frames, data processing will be very slow.

- Occasionally, the cine will appear to have a completely black background (intensity zero). It is not clear why this happens, but different computers can interpret the same cine slightly differently. The lab computer seems less prone to this, but the IDS camera operator computer can sometimes see the same cine, and have this issue.

- If the camera is unplugged before the PCC is closed, it will freeze.

5.2 How to run PhantomStalker/common errors

The PhantomStalker Matlab script replaces both an older labview program which monitored and saved the recorded .cine files to the hardrive, and a python script from NSTX whose job was to convert the .cine files to .mat files and save them to the Temp drive. The new script trims the movies to the shot length, saves them in the tree, and saves the converted .mat file to the network. It is compatible with PDC and regular shots.

To run it, all of the files in `T:\RChandra\Phantom\PhMatlabSDK` must be on the Matlab path, except `\PhMatlabSDK\bin\win32`. That folder must be explicitly excluded.

PhantomStalker requires several precompiled .dlls, which in turn require a compatible C compiler. It has proved extraordinarily difficult to find and install such a compiler, which must be compatible with both the OS and Matlab version. For this reason, if the IDS camera control station must ever be relocated, it is suggested that the physical computer itself be moved along with it.

While the PhantomStalker program can run essentially autonomously once started, it does frequently have difficulty finding the correct shot number. A patch has been added for this, but sometimes the program will need to be restarted a few times to work. The ZAP experiment was apparantly using the same server calls, which may have been the cause of some of the issues. Separately, on PDC shots, the period between shots may be shorter than the time necessary to process the cine data. A fix has been implemented to prevent conversion of frames far after the injector shutoff.

5.3 Overview of Cine File Structure and Access

The Cine file structure is the proprietary file format of the Phantom camera movies. Significant documentation exists for both the the file format itself (Cine File Format) and the Matlab commands to interface with it (Phantom SDK) both in `\PhMatlabSDK\doc\`.

On the lowest level, each .cine file is broken up into the header, which holds all of the information about the state of the camera settings at the time of collection, and the image processing settings that have been applied post hoc, followed by the frames themselves. Inside each frame, each pixel intensity is stored as a 12 bit value, broken up across two bites. Note: there occasionally appears to be a (small) scalar offset on the intensity values, when compared to the PCC program.

On the higher level, the most important concept to keep in mind is the difference between frame time, trigger time, exposure and interval. Frame time is the timestamp associated with the beginning of the frame. Trigger time is the time that has elapsed between the start of the frame in question, and the time that the injectors turned on (or, the time that the server sent out the "PHANTOM_TRIGGER" event). Exposure is how long the shutter was open for, which is the same for all frames in a given cine. Interval is the length of the frame. Importantly, Interval is slightly longer than Exposure because some time is necessary each frame to reset the CCD. Frame rate sets interval, not exposure. For the purpose of comparing to simulation, and producing phase portraits, IDS shifts the frame time so that it is zeroed to the start of the shot, and so that the timestamp is in the middle of the frame. Once turned on, the camera is recording frames continuously, once the trigger is sent out from the server, the camera, which will be in the middle of recording a frame, starts the cine file at the next frame, labels it with frame time zero, and records how long before that time the trigger happened. IDS corrects this by adding the trigger time to the frame time, and 1/2 of the interval time. This is done near the end of PhantomStalker.m

Access of data in the cine can happen several ways. As previously, this used to be accomplished with first a LabView script, and then a python script from NSTX. Currently, it relies on several pre-compiled DLLs, the used of which is tricky at best. The Phantom SDK is well documented, although care should be taken as newer versions of the SDK seem to have slightly different syntax (they are somewhat more object oriented). Note that some functions do not have wrappers built in. See GetCineAuxData.m for an example of this. There are several quirks to data access using the SDK, particularly surrounding the variable types that it uses, and converting them back into primitive types. All of these are well documented, but it can be hard to find.

PhantomStalker.m should have relatively clear examples of how to get a handle to a cine from the camera, and then pass that handle around to various methods to extract data.

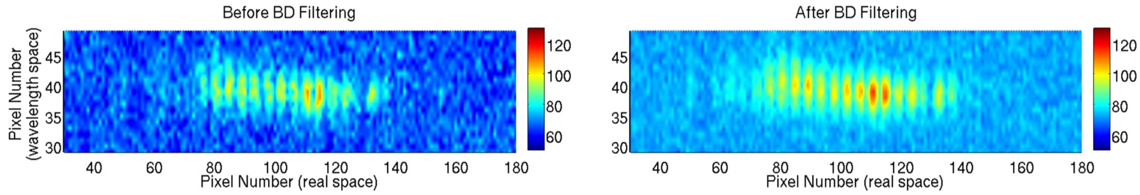

Figure 11: Before and After BF Filtering, HIT-SI Shot 129499

6 Data Processing:

6.1 Stages of Batch_Correct2:

- BD Filtering:

  See Brian Victor paper for the mathematics behind the use of SVD decomposition (Biorthogonal-Decomposition), and Aaron Hossacks Ph.D thesis for IDS applications. For our purposes, it is sufficient to consider it solely a filtering tool. See Fig 11. Note that Dynamic Mode Decomposition may be a future area of filtering exploration.

- How lsqcurvefit finds the gaussian

  Using the calibration values in PEAKS, a window is fit around every fiber, centered on x0,y0, with some window xWing and yWing, set in loadParams, for every frame. In gaussFit2D, extra methods are applied to get the best initial guess possible, such as shifting the window around slightly, and modifying x0,y0 to correct for bumping the spectrometer. Note that these changes are not what the final fits will compare themselves to in calcPhysics. The lsqcurvefit package performs the nonlinear least-squares minimization (using the Levenberg-Marquardt algorithm) of the data onto a trial function (the gaussian), given the initial guess. A slightly modified version checks the error on this fit, an option selected in loadParams as well.

- Calculating Temperature and Velocity:

  The mathematics that underly the conversion of $\Delta\lambda$ to velocity and $\Delta\sigma_\lambda$ to temperature are explained in greater detail in Aaron's thesis. For the purpose of this manual, it is sufficient to know that using PIX_SP we convert Δy to $\Delta\lambda$, and $\Delta\sigma_y$ to $\Delta\sigma_\lambda$, and then apply:

$$v = c\frac{\Delta\lambda}{\lambda} \qquad \left[\frac{m}{s}\right] \tag{1}$$

And:

$$T_i = \frac{\sigma_\lambda^2 c^2 m_i}{\lambda_0^2 k_b} \qquad [eV] \tag{2}$$

6.2 What information Batch Correct takes in, and returns

Batch Correct is the primary data processing tool used by IDS. Each .mat file is loaded in from `T:\PhantomMovies` (although in theory it could be loaded in from the imaging3 tree: this should be tested), and filtered using `BD_Filter`. The processing of the raw data requires it to pull down calibration data from the server (`loadParams`), along with several user-supplied values and modifications. It fits gaussians to each fiber, in each frame, of each movie, and uses the y-width and center to calculate temperature and velocity. There are many places where manual modifications have been added in to improve fitting accuracy, etc, such as the window size around each gaussian, and values having to do with modifications on the calibration data. It also loads in additional information from the server about the injectors.

Batch Correct saves a file named "dat[shotname]" to `T:\IDS\Data_Repository\`. This file contains:

- The calibration data from Sec.3.2.
- A matrix of celocity data, organized by index×time, where PEAKS converts from index to channel number and impact parameter.
- A matrix of temperature data, organized like the above.
- A matrix of intensity data, organized like the above.
- The timebase for the above.
- The injector currents, and the injector timebase (for all injectors).

- The quadrature toroidal current, stored in `I_TORR`.
- The high and low error estimates, if the `calcError` flag was set in `calcParams`.
- The filtered data from the movie.
- The impact parameters (also stored in the calibration data).
- The shot name and number.

7 Data Analysis and Visualization

7.1 `Compare_Plots3_MultiLine`

This plotting routine (Fig 12a), frequently used first as a sanity check on the data, produces the color square surface plots seen on some posters in the hall. It can display the analyzed quantities stored in the .dat file, and do some analysis on its own. Which fiber array is being plotted can be changed, as can whether the data is displayed by impact parameter or channel number, and time point or time point number. While the surface plot style may not be as useful to the uninitiated, with some practice it is a quick and easy way to check if the data overall looks reasonable. The "`_Multiline`" version of Compare_Plots3 is for simultaneously plotting the two fiber bundles in the Dual Mohawk HIT-SI3 configuration (Fig 8). However, it still can plot a single array, based on the `torPlot` parameter.

7.2 `MultiPlot`

This is a slightly newer routine (Fig 12b), which plots the one dimensional (usually velocity) data for some range of channels, offset by a constant value. It has an extensive analysis option, which can perform an FFT, reconstruct the data if it meets some threshold for injector frequency component, and then report the phase of each channel, as well as the net velocity difference between fiber arrays (net toroidal flow, if the fibers are in upper/lower mohawk 6), and the average integral over a half cycle (the displacement). The

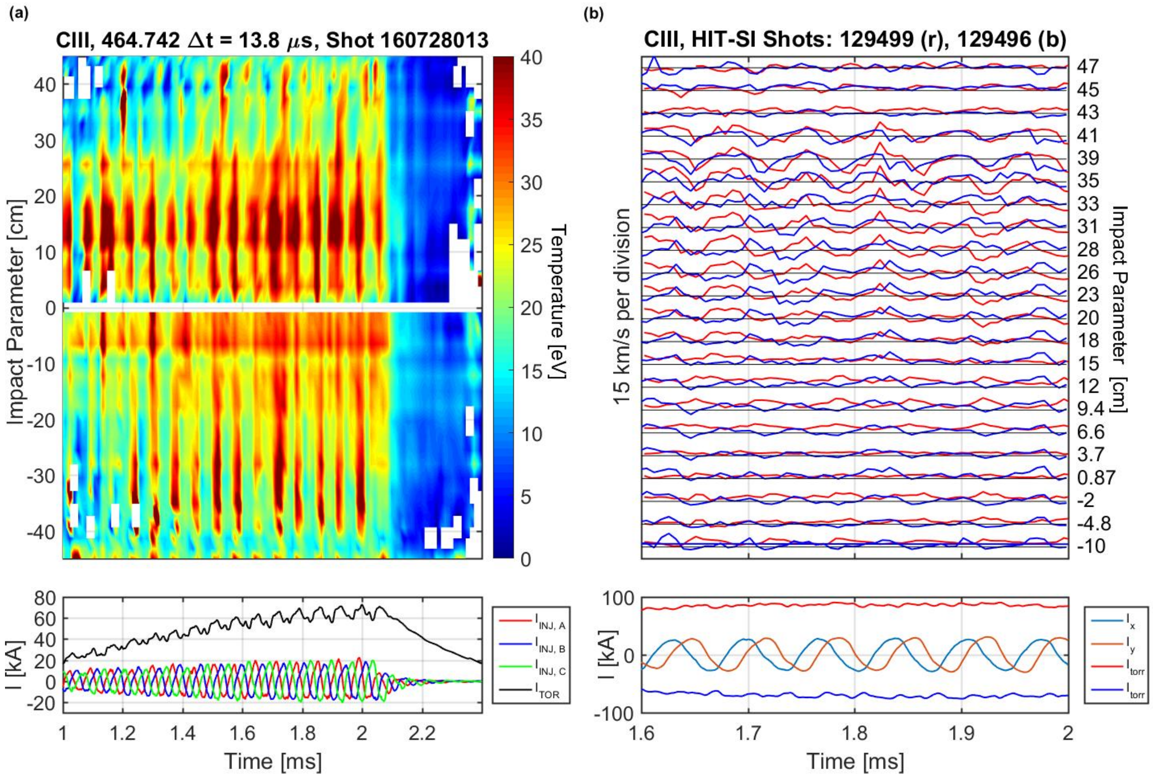

Figure 12: (a): Temperature from HIT-SI3 shot 160728013 C III line. (b): Velocity from HIT-SI shots 129499 (red) and 129496 (blue), C III line. Black lines mark zero velocity for each chord. Data is missing where a gaussian could not be fit.

phase between channels, and between fibers can give interesting results about the level of coherent motion. See Aaron's thesis for a partial description of this phenomena.

7.3 IDS_IAEA

This code is a more compact visualization of the analyzed profiles that are produces by MultiPlot. The code was written by Aaron for the IAEA Nuclear Fusion paper hence the name. It has the advantage of being able to compare two datasets simultaneously. In Fig 13, HIT-SI and HIT-SI3 are compared for the same carbon line.

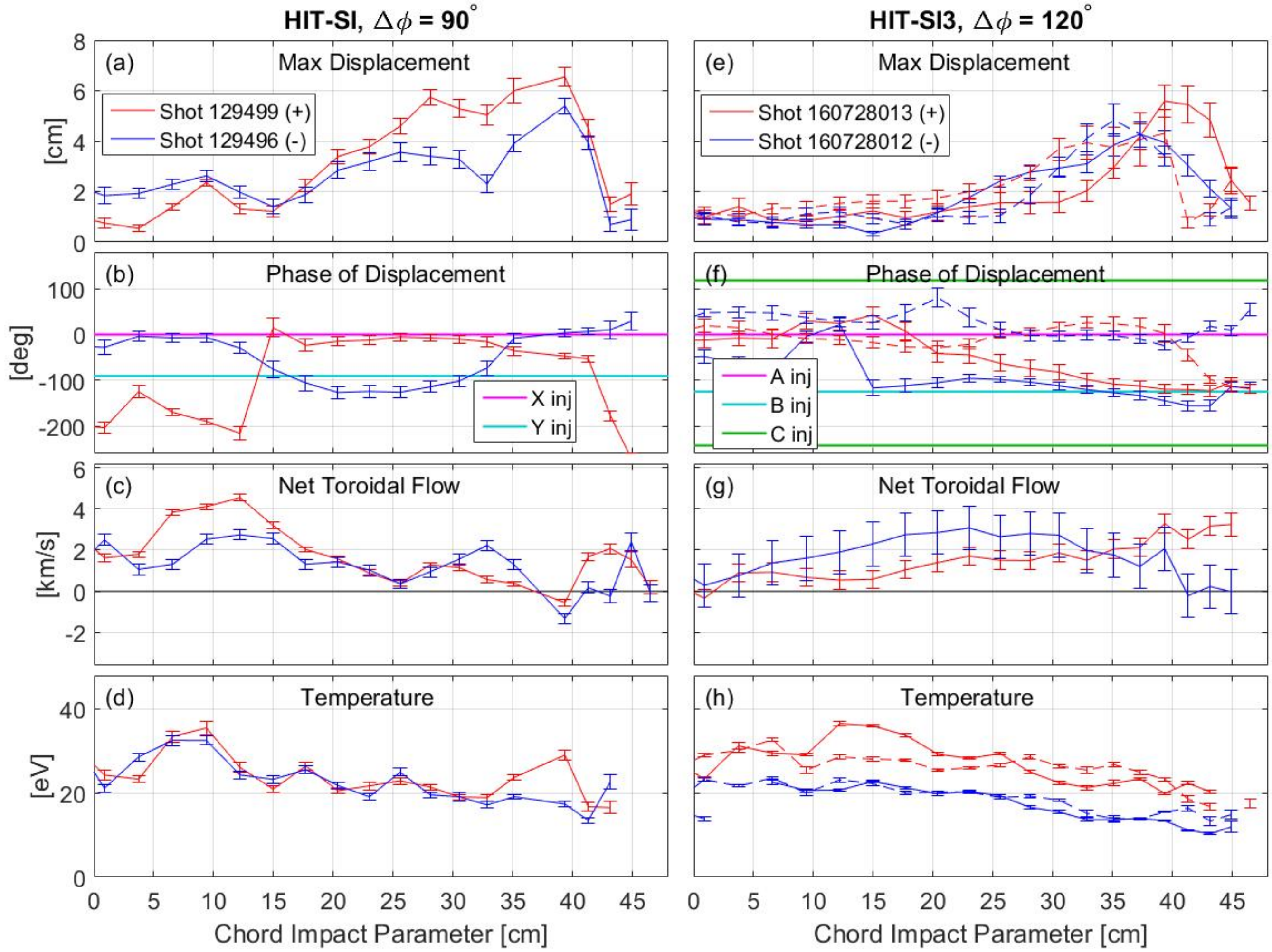

Figure 13: All Analyzed Profiles, HIT-SI vs HIT-SI3 for CIII Line

8 Issues with processing and comparing to NIMROD and PSI-TET

Comparison with numeric simulation is a key role for cross validation of IDS. Below is a partial list of some of the issues to be aware of when attempting this.

- It is often necessary to shift the simulation timebase to make the toroidal current match that of the experiment as close as possible. This can present issues.
- Before NIMROD implemented a synthetic IDS output, the line integration was performed in VisIt. This is time and resource consuming, and should be avoided when possible.
- `Calc_Geometry_5` has an output to give the start and end positions of the light-chords for NIMROD. These should be visually verified, to confirm that they match machine coordinates, before use.
- The symthetic data must be run through the same analysis software, for consistancy purposes. This can present issues, because, for example, there may not be tree data for the synthetic shot.

9 IDS-Relevant Research Questions:

- Toroidal/Poloidal Flow Profile
- Velocity zeroing using opposing ports
- Complete flow profile using upper/lower mohawk 6
- Phase profile by impact and velocity profile by impact over time
- Peak Flow scaling (by toroidal current, injector phasing)
- Velocity with fibers in opposing ports
- Comparing anything to simulation