BES Tutorial

DISCLAIMER:

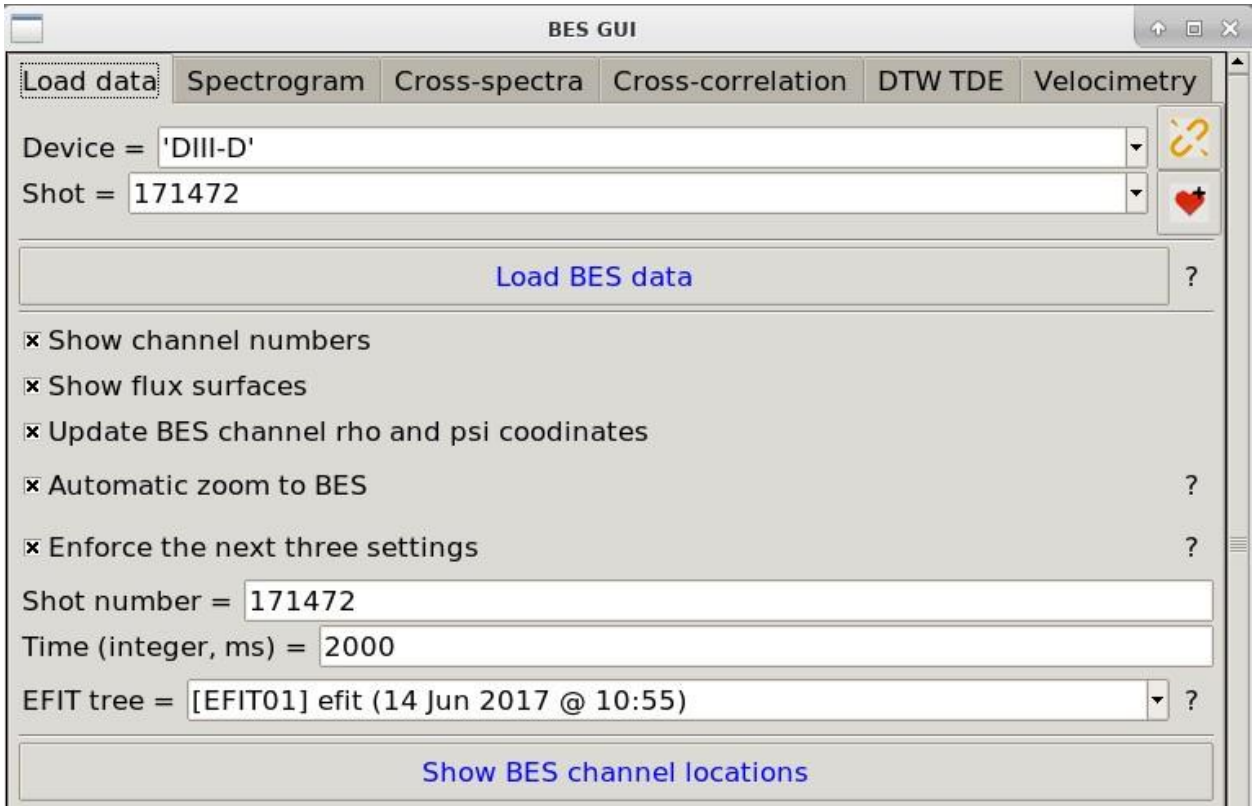
This is an editable public document! When you make a change to it, it will be visible immediately from everyone! Feel free to edit it and help other OMFIT users!

Since this is an evolving document, there may be some small inconsistencies as different figures have been taken by different people with different versions of OMFIT for different analyses.

Introduction

This document describes how to analyze density fluctuation data obtained with the beam emission spectroscopy (BES) diagnostic using the OMFIT BES module.

Workflow



The 'Load data' tab.

Start at the *Load data* tab, enter the device and shot you want to analyze BES data for, then click *Load BES data*. Raw data can only be loaded for a single shot at a time due to the large memory footprint of BES data (multiple GBs per shot). Processed BES data is preserved in the OMFIT tree so you can analyze BES data from one shot, then load data from a different shot and still retain the old analyzed data.

Clicking *Load BES data* does not actually load all the raw BES data, but instead creates links to the MDSplus tree locations where the BES data is stored. This helps avoid long load times and high memory usage. The raw BES data is not actually loaded into memory until it is required for an analysis.

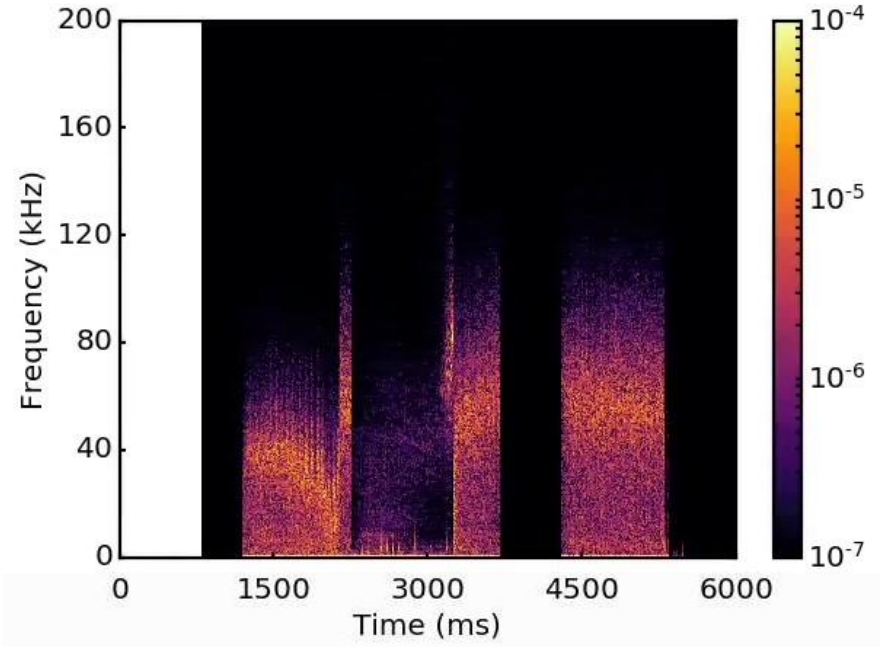
Underneath the *Load BES data* button are options for making an RZ plot of the BES channel layout. If the EFIT settings are entered then the flux surfaces are also plotted, allowing you to quickly see the location of the BES channels within the plasma.

Once BES data is loaded, different analyses can be performed using the various tabs in the GUI. A typically order for analyses is to first look at spectrograms to see the turbulence's frequency range and how it evolves through the discharge. Then use cross-spectra to look at detailed power, phase, and coherence spectra, as well as density fluctuation amplitudes. The cross-spectra information can then be used to perform more sophisticated analyses including turbulence parameter estimation (group velocity, correlation length, decorrelation time), dynamic time warping (DTW), or velocimetry.

Analysis Options

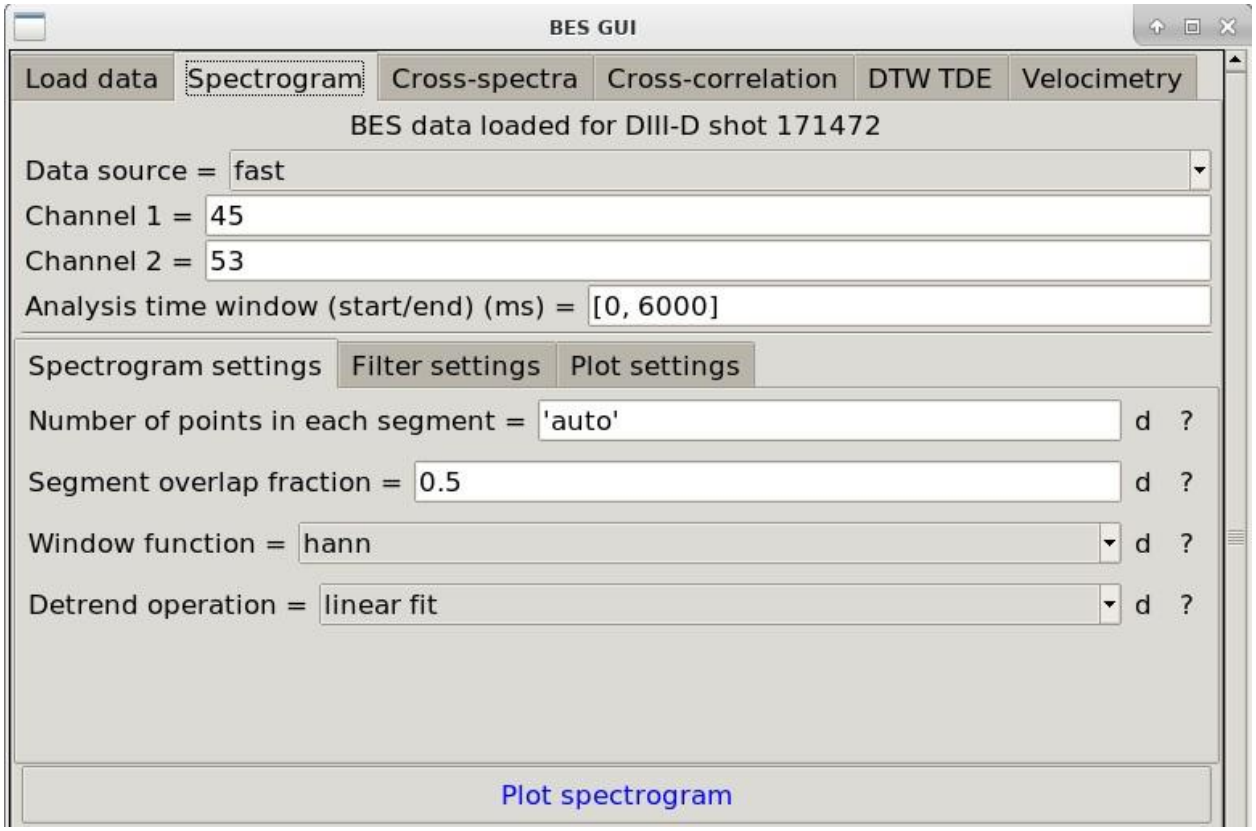
Spectrogram

Spectrograms give a visualization of the turbulence spectrum as it evolves in time throughout the discharge. They are often the first quantity you want to look at when analyzing BES data. For example, the turbulence spectrum below shows turbulence over the 0-80 kHz band during the L-mode period from t=1200-2250 that is then suppressed during the H-mode period from 2250 to 3250 ms, only to reappear again after the H-L back-transition at 3250 ms. In particular, spectrograms are useful for identifying time periods where the turbulence is stationary, which are then used to perform analyses (such as cross-spectra or cross-correlations) that average over this time window.



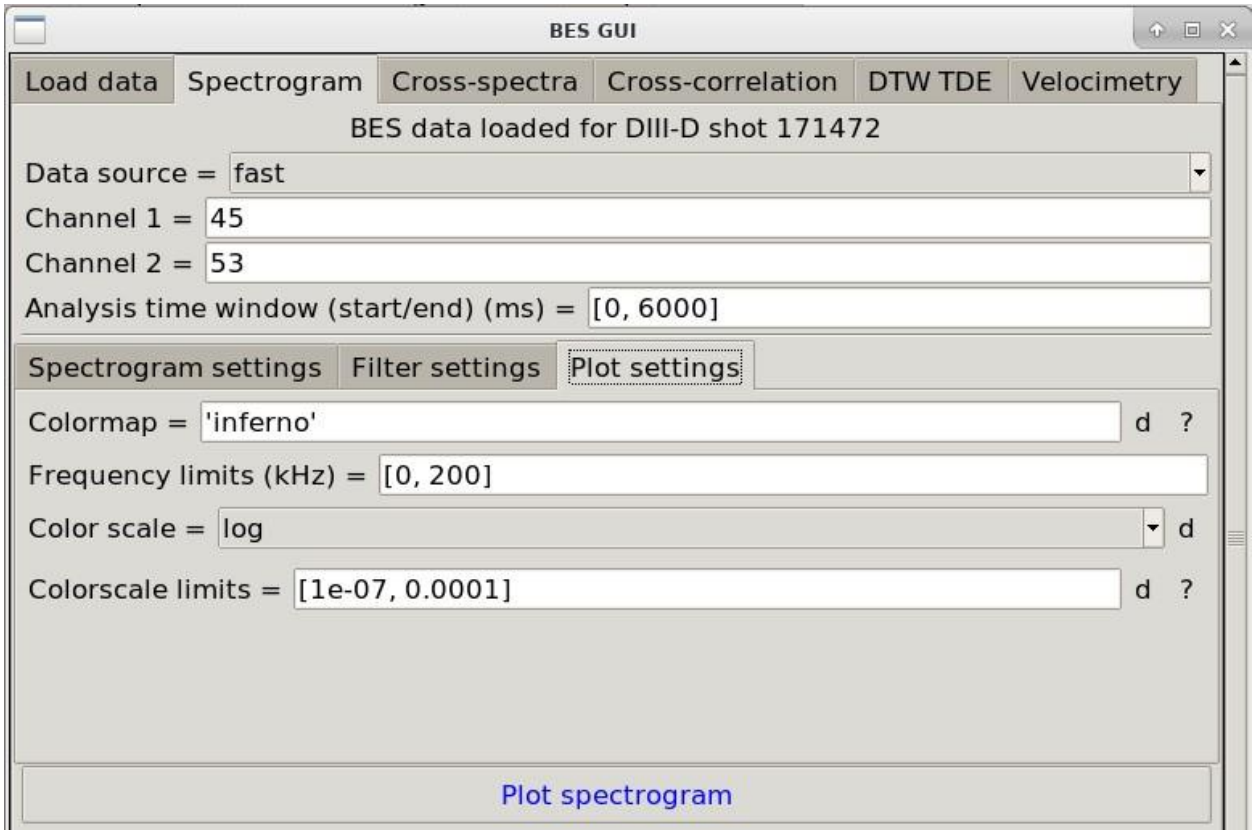
Example spectrogram of BES channels 45 & 53 from DIII-D shot 171472. The significant reduction in turbulence during the H-mode phase (t=2250-3250 ms) is clearly visible.

To plot a spectrogram first select the pair of channels and time window to use. Typically, a pair of adjacent, poloidally-separated channel at the same R are used. Or the same channel number can be entered twice to produce an autospectrum. The *spectrograms settings* sub-tab lets you set options for the FFT computation used to produce the spectrum. The default values should work fine for most cases.



The 'Spectrogram' tab and 'Spectrogram settings' sub-tab.

Visual aspects of the spectrogram can be tweaked on the *Plot settings* sub-tab. The popular 'Standard Gamma-II' colormap used by the IDL Inspect code is available but is discouraged due to its poor perceptual uniformity (see https://www.youtube.com/watch?v=xAoljeRJ3lU for more info on problems with many popular colormaps). The default *inferno* colormap (or some other perceptually uniform colormap) is recommended. Unfortunately, the colorscale limits must often be tweaked manually to produce a good looking spectrogram. They can be set to *None* to simply use the min and the max of the spectrogram but this often produces poor results.



The 'Spectrogram' tab and 'Plot Settings' sub-tab.

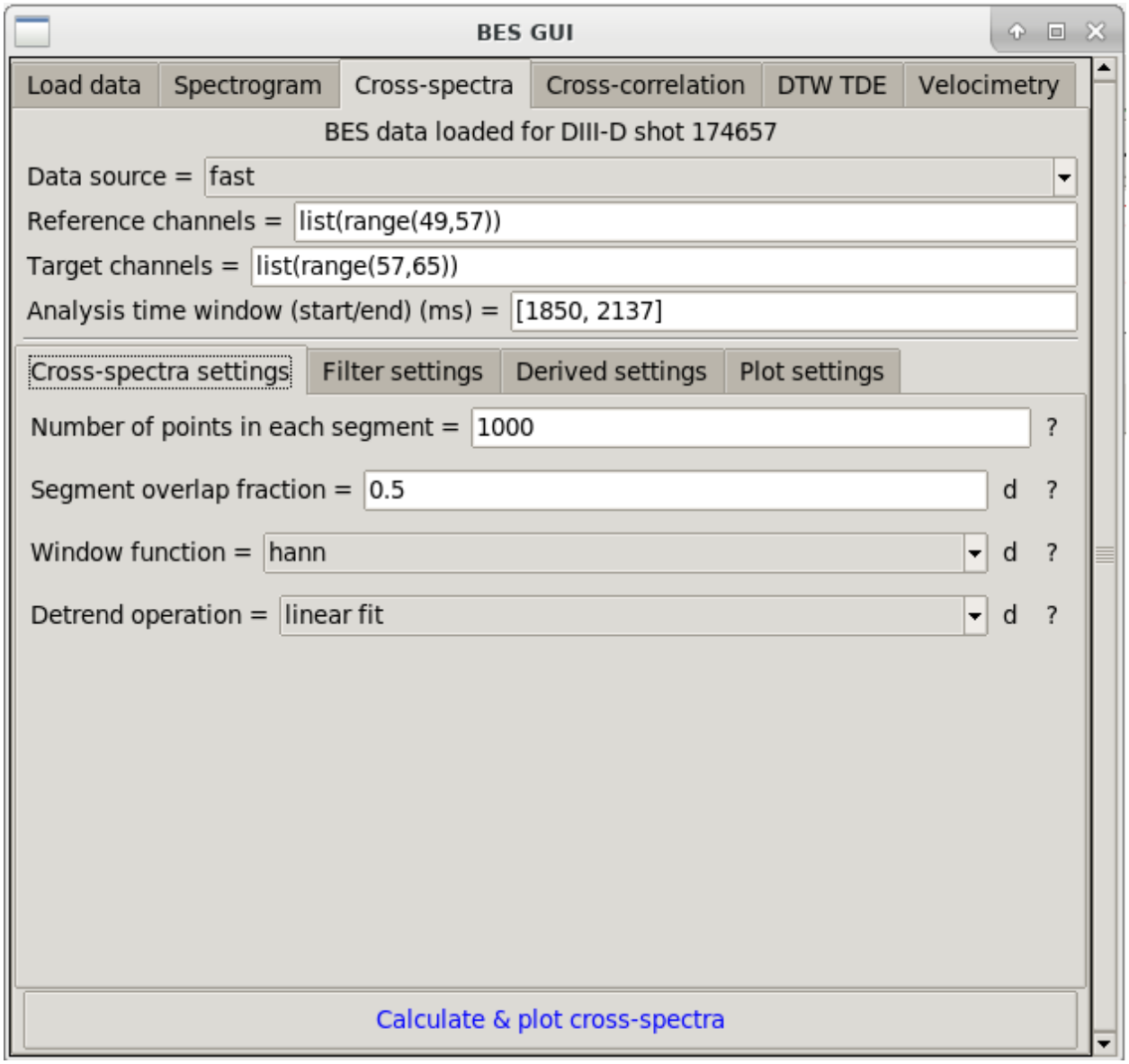
Cross-spectra

The cross-spectra analysis tab allows you to compute frequency-domain quantities between multiple reference and target channels, for multiple time ranges within a single shot. This is also your entry point for computing the frequency-integrated fluctuation power. The cross-power between two adjacent channels rejects channel-specific noise power (due to the detector electronics, photon discreteness, etc) which is not correlated between the detectors of distinct channels, while keeping fluctuation power from the plasma which is correlated over the distance between channels. For this reason, the cross-power between poloidally-adjacent BES channels is generally used to compute the fluctuation amplitude, by integrating the fluctuation power over the frequency range of interest. (The poloidal correlation length of fluctuations is generally much longer than the channel-to-channel spacing.)

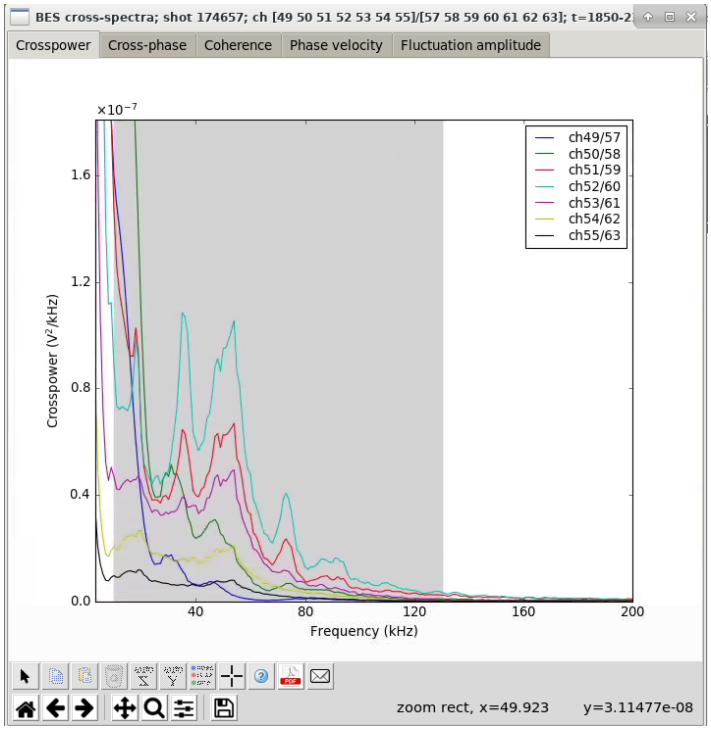
The underlying data processing relies on the scipy.signal.csd method.

The reference channels and target channels between which you want to compute the cross-spectra are supplied as length-matched lists. (You can 'cheat' and supply a single integer as the reference channel value, which will be the default reference channel for all the target channels supplied.) Interchanging the reference and target channels just reverses the sign of the cross-phase, but everything else should stay the same.

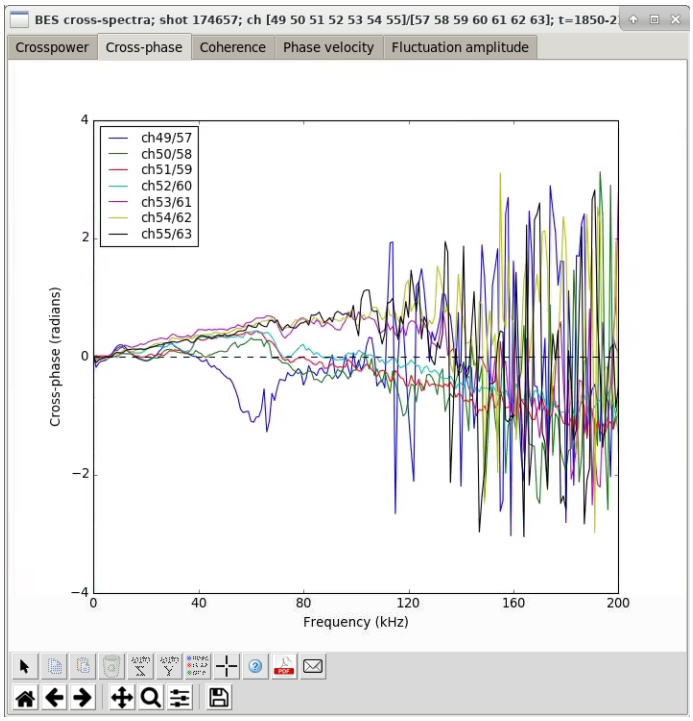
You supply an analysis time window as a list of [start,stop] times. You can supply multiple time windows as a list of lists: [[start1,stop1], [start2,stop2], etc]. Each time window will bring up a new plot window, so you may want to limit the number of time windows to a dozen or so. The time window is broken up into 'segments' containing a specified number of time points (ex: in DIII-D, the time basis of the digitizer is 1us, so 1000 points is 1ms). Each segment is Fourier-transformed, and the resulting cross-spectra from each segment are averaged together to yield the final result. The default window function 'hann' is used to smooth the resulting spectrogram. Increasing the segment overlap fraction also acts to interpolate the spectrum. The detrend option removes low-frequency trends in the data.



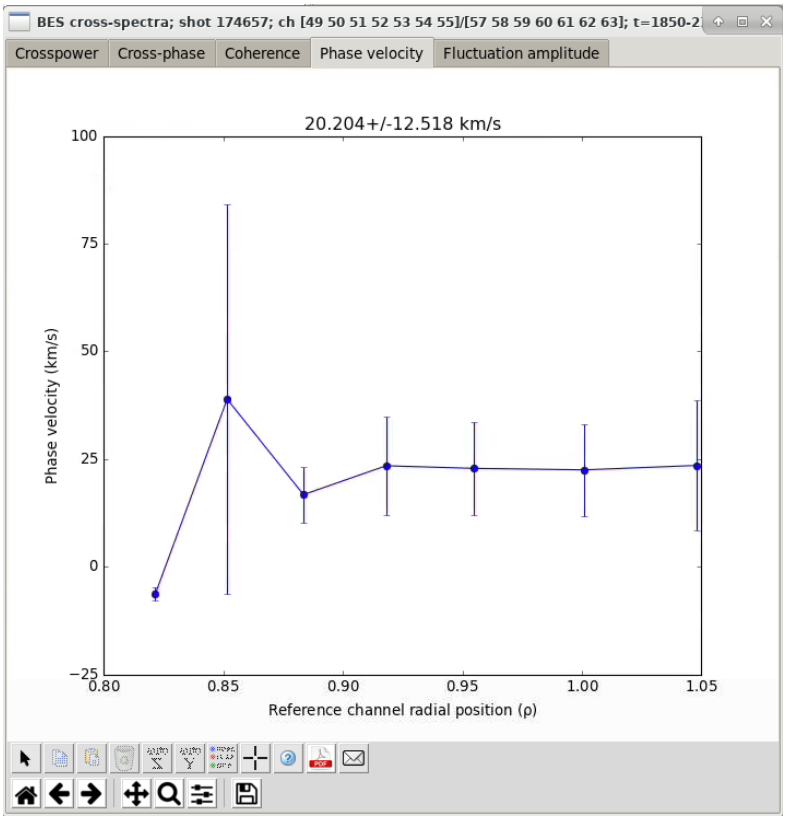
Cross-spectral analysis tab, 'Cross-spectra settings' sub-tab



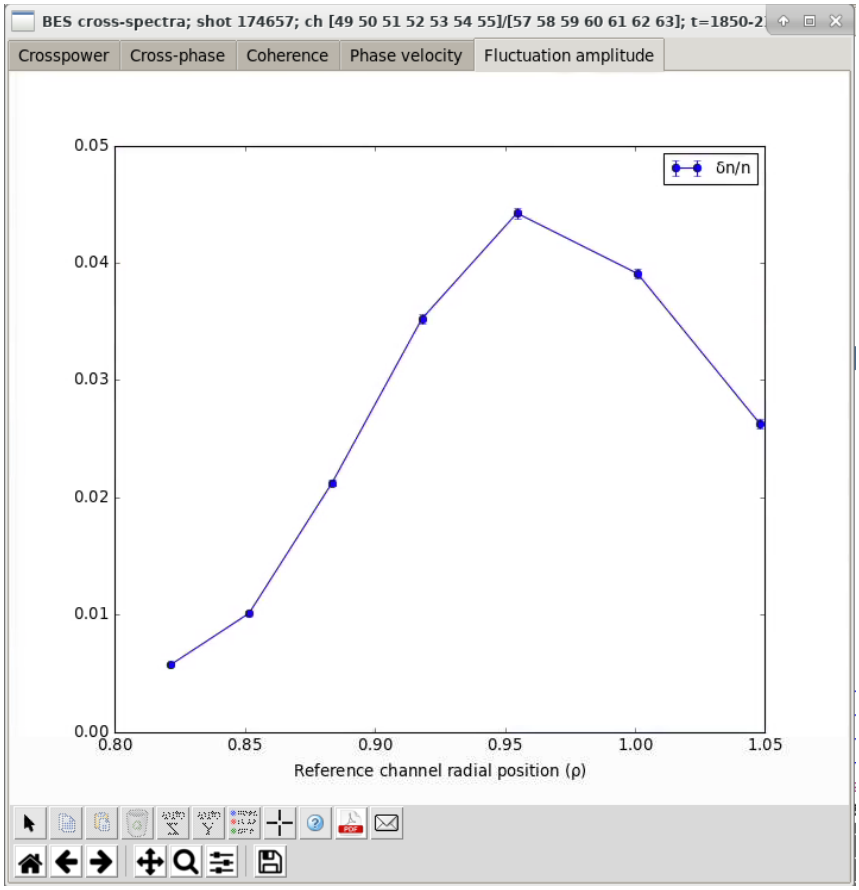
Linear-scale cross-power. The grey shading indicates the frequency range over which the fluctuation amplitude is being computed. Whether this shading is displayed can be toggled on the 'plot settings' sub-tab.



Cross-phase tab.



Phase velocity is computed from the slope of the phase over the analysis frequency range, which yields a time delay value between the channels. This is used to compute v=x/t.



Relative fluctuation amplitude as a function of radial coordinate rho. The fluctuation amplitude peaks just inside the last closed flux surface at rho=1.0, which is typical.

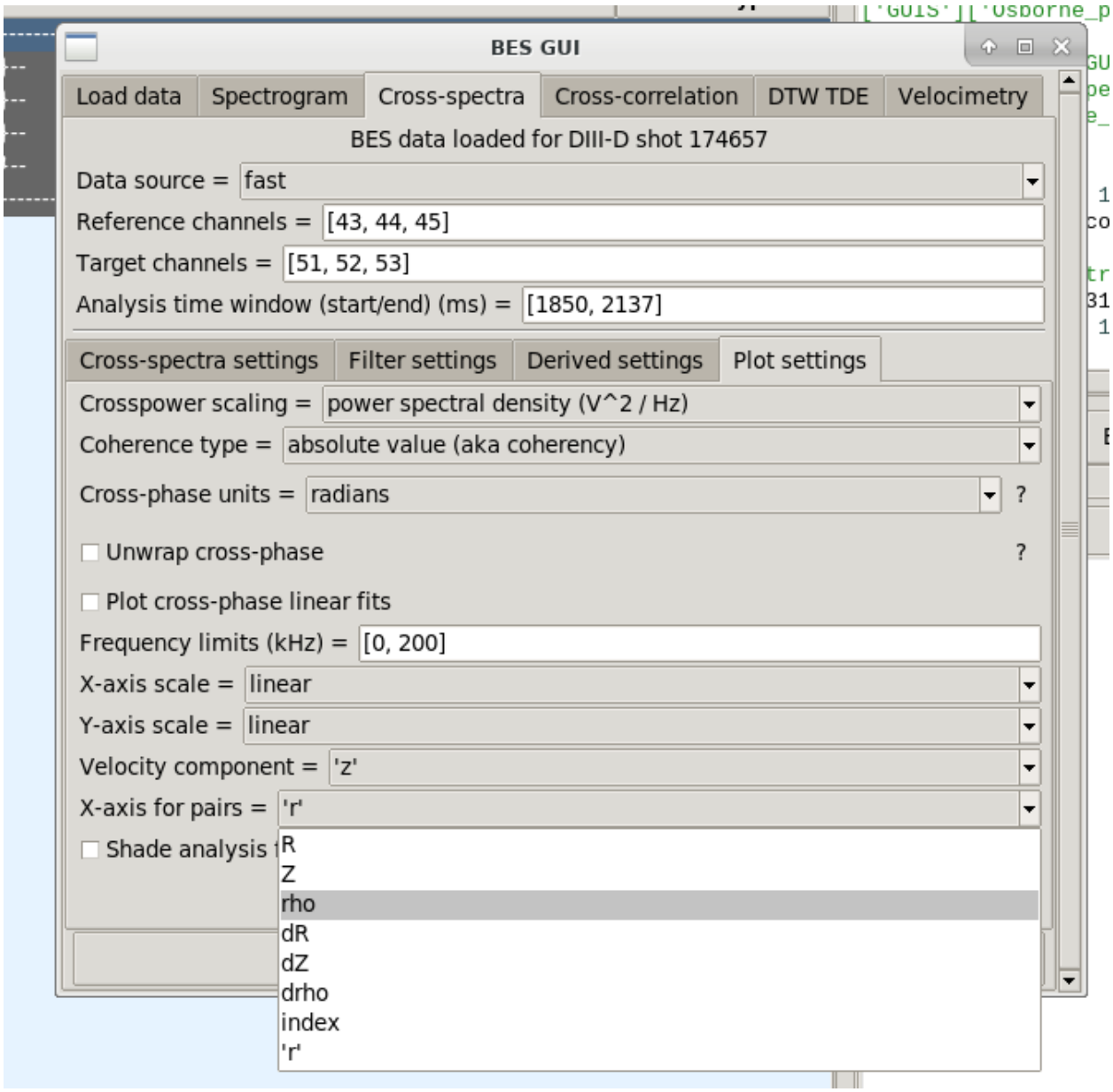
The appearance of the plots can be customized to an extent using the 'Plot settings' sub-tab. The normalization of the cross-power, the scaling (linear or quadratic) of the coherence, and the units of the cross-phase (radians/degrees) can be selected. Unwrapping the cross-phase can be useful to remove discontinuities in the phase plot. The linear fitting method used to find the derivative of the phase with respect to frequency (for computing the phase velocity) can be plotted using the toggle here.

The default frequency range can be set here, as well as selecting log/linear scales for the vertical & horizontal axes.

The phase velocity is computed either from the vertical (Z) or horizontal (R) distance between the reference and target channel in each pair, along with the time-lag estimated from the slope of the phase-vs-frequency curve. There is a toggle here to select which component should be plotted.

For the derived quantities, the horizontal axis is a coordinate that tracks the 'pair' dimension (as in, reference/target pair number). The pair number (index) is always available, as are the R & Z

coordinate of the reference channel in each pair. The ‘rho’ coordinate will be available if an EFIT was loaded at the ‘load data’ tab -- the ‘rho’ values are computed for the R,Z coordinates of each channel based on the EFIT time slice that was loaded when the cross-spectra was computed. The ‘dR’, ‘dZ,’ and ‘drho’ options specify the difference in the respective coordinates going from reference to target channel.



Selecting the horizontal axis coordinate for the derived-data plots (eg, fluctuation amplitude or phase velocity vs rho).

Cross-correlation

Dynamic time warping (DTW) time-delay estimation (TDE)

Velocimetry

Filter settings

Datasource

Analysis FAQs

- What is the origin of the large peak in the cross/auto-power spectra below about 10kHz in my DIII-D data?
    - This is interference/noise originating from the neutral beam intensity rather than the local electron density fluctuations. The neutral beam sources produce some low-frequency fluctuations that are coherent across the entire beam. Also, there are fluctuations in intensity due to fluctuating levels of attenuation of the beam as it passes through the pedestal/scrape-off-layer; this is referred to as ‘shadowing.’ These fluctuations are well-correlated in the major-radial-direction, and may have finite correlation length and phase velocity in the vertical (poloidal) direction due to the edge turbulence properties.
- How do I tell what the ion/electron diamagnetic directions are for the fluctuation phase velocities?
-

Regression Tests

Regression tests should ideally be run every time a change is made to the BES module code, and are required for any code changes that are pushed to the public version of OMFIT on the gafusion repository.

Regression tests are located at OMFIT-source/regression/test_BES.py The beginning of the script has lines that turn on and off the various regression tests. For small code changes, only

the relevant regression tests need to be run. For code changes that will be pushed to the public version of OMFIT the full suite of regression tests should be run.

To run the regression tests use the command `omfit --cwd -g test_BES.py` (for tests involving GUI elements) or `omfit --cwd test_BES.py` (for tests without any GUI elements). This will open OMFIT and run all the tests. Any errors that occur will be logged in the OMFIT console.

Developer Info