

NeuroSpec 1.0 User Guide

Introduction.

This guide provides an overview of the **NeuroSpec** statistical signal processing archive version 1.0. The archive consists of a number of MATLAB functions for performing a Fourier analysis of time series and/or point process data, and plotting the results. Included in the archive is a demonstration script, see below.

Software Requirements.

These routines require the MATLAB system to be installed. You will require **version 5.0 or higher** to run these routines. They have been tested using the PC versions 5.3 and 6.1. No additional toolboxes are required, only the base system is needed. If you have an older version of MATLAB please read the FAQ on the neurospec web site.

Installation.

To install the archive:

1. Download and Unzip the file `NeuroSpec10.zip`
2. Either:
 - Install the two directories `NS_spect` and `NS_plot` in the MATLAB path, or
 - Copy the files in these two directories to your current working directory for MATLAB.

Format of Data.

NeuroSpec can process either time series or point process (spike train) data. Below is specified the format the data should be in for processing by these routines. This format should readily accommodate most types of experimental/simulated data. **N.B.:** Failure to specify input data in this format may give incorrect results. We adopt the MATLAB convention of specifying multivariate data in column format.

Time Series data should be in column format. Both single column and 2D matrices can be processed. See the script `sp2_demo1.m` for an example of processing matrix data. Single column data passed to the **NeuroSpec** routines should have the same number of data points in each column. Time series data is assumed to represent regularly spaced samples taken from simultaneously observed signals, with a known sampling rate. Integer or real values can be processed. The sampling rate has to be input to the analysis – this, in conjunction with the FFT segment length, defines the frequency resolution of the analysis.

Point Process (Spike Train) data should be in column format. Since it is unlikely that two or more spike trains will have identical numbers of spikes (or events), these will generally be stored in separate variables. Each row specifies the time of a spike as an integer number of the sampling rate. Real valued data should first be converted to integer values using an appropriate sampling rate. If analysing hybrid data, spike trains should be specified with the same sampling rates as the time series data. The spikes times should be specified in ascending order of occurrence.

What is included in the archive.

The core of the archive is the routine `sp2_fn2.m`. This routine implements a periodogram based spectral estimation for two channels of stationary data. It estimates the two auto spectra, coherence and phase in the frequency domain, and the cumulant density in the time domain. The routine `sp2_fn2.m` is not designed to be called directly from within MATLAB. Three routines provide the interface between your raw data and `sp2_fn2.m`. These functions are:

1. `sp2a2_m.m` For processing two time series.
2. `sp2a_m.m` For processing hybrid data (one time series and one spike train).
3. `sp2_m.m` For processing two spike trains.

Also included are a number of routines to plot the results of a spectral analysis. The interface to these routines is provided by the function: `psp2.m`

Using the routines.

This section gives an overview of the syntax and arguments, including the use of options for the `sp2*` functions in **NeuroSpec 1.0**. Supposing you have in MATLAB the following variables:

```
dat1, dat2: two time series of equal length
sp1, sp2: two spike trains specified using the same sampling interval as the time series data.
```

Suppose the sampling rate is 1000/sec; in MATLAB: `rate=1000;`

Define the segment length for the FFT as 2^{10} (1024 points); in MATLAB: `seg_pwr=10;` (Note: the segment length is always taken as a power of 2, for the `sp2*` functions it is specified as this power)

Using the routines: Time series analysis.

To undertake a Fourier analysis of the two time series data use the command:

```
[f,t,c1]=sp2a2_m(dat1,dat2,rate,seg_pwr);
```

Options are included by specifying an option list in a text string. For example, if you wish to perform a linear de-trend on the data before analysis use:

```
[f,t,c1]=sp2a2_m(dat1,dat2,rate,seg_pwr,'t2');
```

If you wish to rectify both channels and then perform a linear de-trend prior to analysis use:

```
[f,t,c1]=sp2a2_m(dat1,dat2,rate,seg_pwr,'t2 r2');
```

For further details on the use of options, see the file `sp2a2_m.m`. The order in which the options are specified is not important, however, only a single options string should be used, multiple options should be specified with a blank space between. The `'i'` option inverts the channel reference order for the phase and cumulant density estimates. In the above estimate `dat1` is assumed to be the reference (input) signal and `dat2` the response (output) signal. Setting the `'i'` option inverts this so that `dat2` is assumed to be the input signal. The same effect can be achieved by swapping the order the data vectors are passed to the routine, however this is not possible for hybrid analysis (see below). The `'m'` option applies a simple filter to suppress the effects of mains, it is perhaps most useful to suppress any 20 ms oscillations in time domain estimates resulting from 50 Hz

contamination of signals. This can be set to 60 Hz if necessary in the MATLAB function `sp2_fn2.m`, using the variable `line_freq`.

Using the routines: Point Process (Spike Train) Analysis.

If you wish to undertake a Fourier analysis of the two spike trains use the command:

```
[f,t,cl]=sp2_m(sp1,sp2,rate,seg_pwr);
```

The options available in `sp2_m` are 'i' and 'm', they work as specified above.

Using the routines: Hybrid (Time Series/Point Process) Data Analysis.

If you wish to undertake a Fourier analysis of the hybrid data `sp1` and `dat2` then use the command:

```
[f,t,cl]=sp2a_m(sp1,dat2,rate,seg_pwr);
```

The spike data should be passed as the first parameter, the time series as the second parameter. The routine assumes the spike train is the reference signal for phase and cumulant density estimates, to invert this the 'i' option has to be used. Rectification and de-trending are only available for time series data, to specify these use options with no arguments: 't r'.

Using the routines: Output from analysis.

The output from the three functions has the same format in all cases, the functions return three variables:

1. `f`: matrix of frequency domain results.
2. `t`: matrix of time domain results.
3. `cl`: structure containing scalar values associated with the analysis including confidence limits (thus the use of `cl`).

For further details and the format of these variables see the `sp2* m` files.

Using the routines: Plotting of results.

Plotting of the output from the `sp2*` files is done using the function `psp2.m`. Assuming that `[f,t,cl]` are returned, then the results can be plotted using a command like:

```
psp2(f,t,cl,100,200,100,0.5,'Good data 1')
```

For a description of the arguments see the file `psp2.m`. The parameters controlling the frequency range and lag range for plots should take note of the nyquist frequency for the analysis and the periodogram segment length which determine the maximum frequency and lag range available. Passing values to `psp2` which exceed these limits will generate an error message.

Reference.

Further details about the multivariate Fourier framework for analysis of time series and spike train data are in the following reference:

Halliday D.M., Rosenberg J.R., Amjad A.M., Breeze P., Conway B.A. & Farmer S.F. “A framework for the analysis of mixed time series/point process data - Theory and application to the study of physiological tremor, single motor unit discharges and electromyograms.” *Progress in Biophysics and molecular Biology*, **64**, 237-278, 1995.

Please consider referencing this article in any scholarly publications resulting from the use of **NeuroSpec**. This article, and other related articles on the Fourier analysis of Neural data are available from David Halliday’s publication page, which can be found at: <http://www-users.york.ac.uk/~dh20/>

Licensing.

NeuroSpec is free software. It is covered by the GNU General Public Licence, a copy of which is included in the distribution file.

Getting Started – using the demo script.

A script `sp2_demo1.m` is included as part of the software archive. This script is intended to demonstrate the use of the analysis functions on artificially generated time series data. Copy the script `sp2_demo1.m` to your current working directory and run it.

In the MATLAB command window type: `sp2_demo1`

A series of 6 plots should be produced. These are reproduced below along with a brief description of what each plot shows. On the following pages are shown plots which have been generated by the demonstration script `sp2_demo1.m`. Note that since they are based around analysis of data generated randomly in MATLAB the precise form of the plots you generate may differ from those shown here. However, the general form of them should be similar. They are provided as a guide to getting started with **NeuroSpec**, included are some comments about interpretation.

Figure 1, below, shows an analysis of two uncorrelated signals, generated from a normal distribution. These can be considered as an approximation to white noise, and thus have a flat power spectrum over the entire frequency range. Most of the coherence values are below the 95% confidence limit, thus no significant correlation is evident. In this case the phase has no valid interpretation, and is randomly distributed. The time domain cumulant density estimate fluctuates about the null value, zero, with most values inside the upper and lower 95% confidence limits, supporting the interpretation of uncorrelated data.

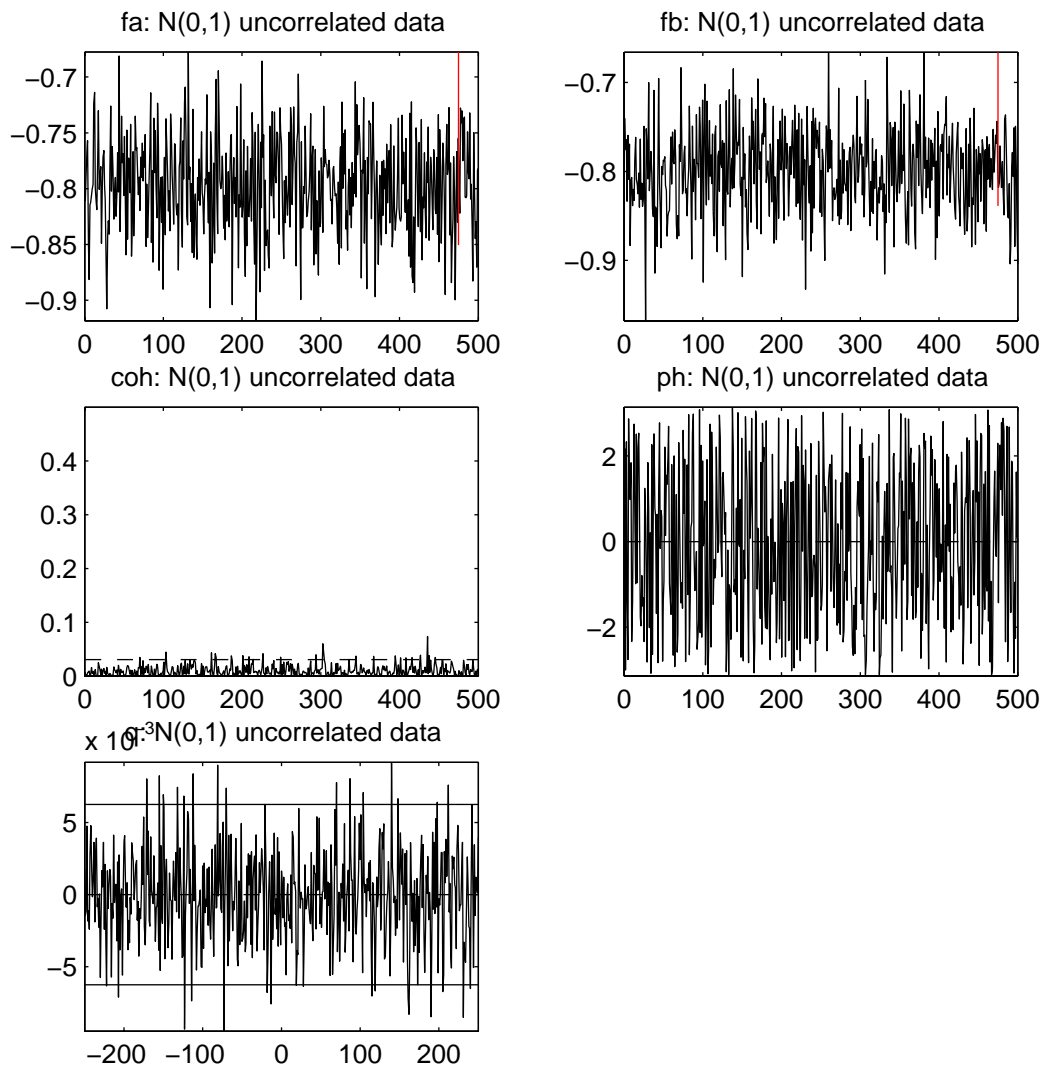


Figure 1. Output from `sp2a2_m`, for analysis of uncorrelated white noise. Top two graphs are plots of the power spectrum of each signal (plotted as \log_{10}), for channel 1 (left) and channel 2 (right). The magnitude of the 95% confidence limit is shown as the red bar in the top right of each plot. The centre two plots show the coherence (left) and phase (right) estimates. The horizontal dashed line in the coherence estimate is the upper 95% confidence limit based on the assumption of independence. The abscissa in the top four plots is labelled in Hz (up to the presumed nyquist frequency of 500 Hz). The lower plot shows the cumulant density estimate, plotted again time lag in ms, can be read a samples in this case. The horizontal dashed line at zero indicates the null value, the solid horizontal lines indicate the upper and lower 95% confidence limits based on the assumption of independence.

In figure 2 we have correlated data, generated by combining white noise signals sample by sample. In this case the coherence is constant around 0.5, indicating that channel 1 can account for around 50% of the variability in channel 2 across the entire frequency range. The phase is constant around 0 radians, indicating synchronous signals, and the cumulant has a large peak at time zero indicating that the time span of the correlation is very narrow, as we would expect for point wise combination of white noise signals (a property of white noise is that values in adjacent time bins are independent).

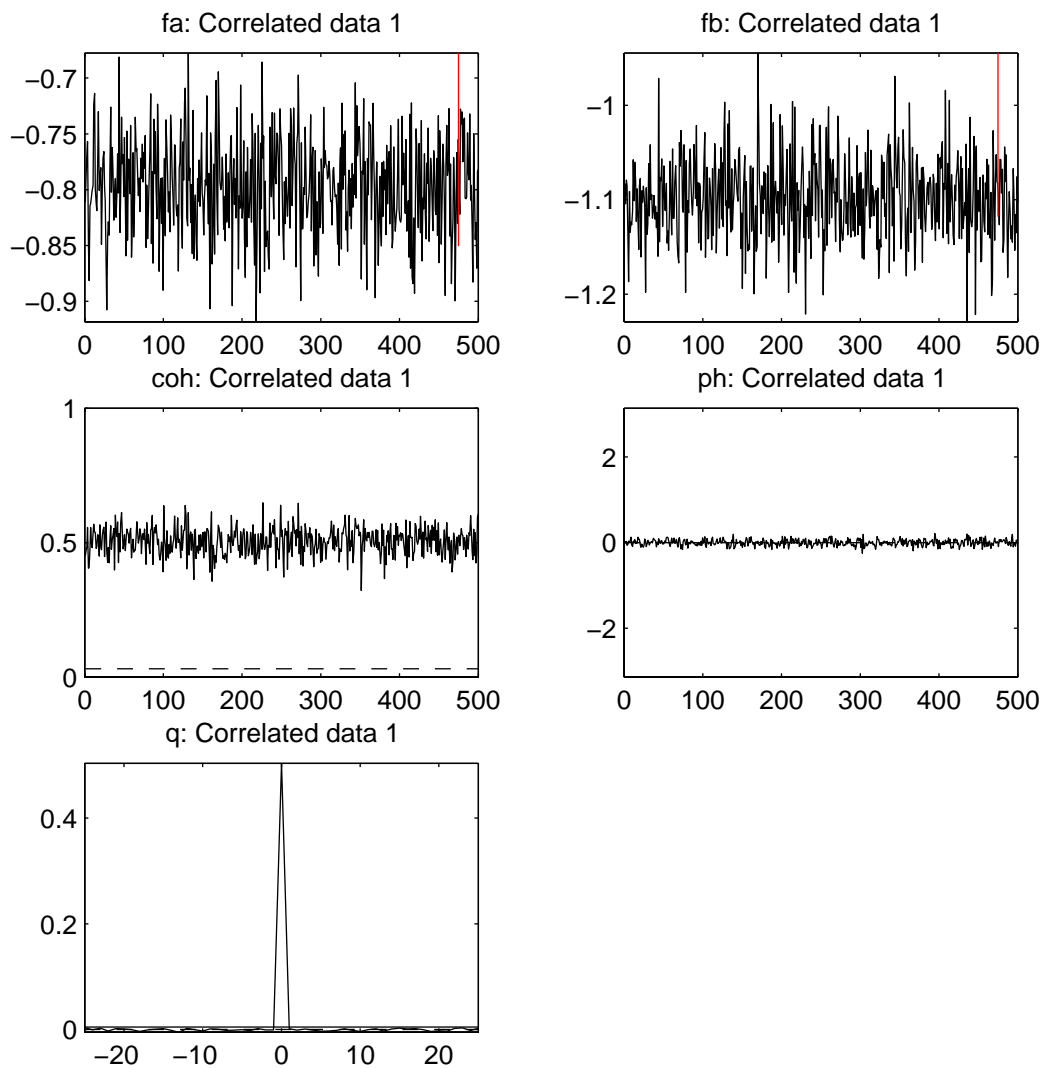


Figure 2. Output from `sp2a2_m`, for analysis of correlated signals. Layout as in Figure 1.

Figure 3 shows the results of introducing a delay of 10 samples (which equates to 10 ms assuming 1 ms sampling) in the correlated data analysed above. The phase and cumulant density estimates change to reflect this delay, the phase has constant slope (equal to the delay) and the peak in the cumulant is now moved to +10 ms lag. The phase estimate is constrained over the range $\pm\pi$, a discontinuity appears as the phase cycles through this range.

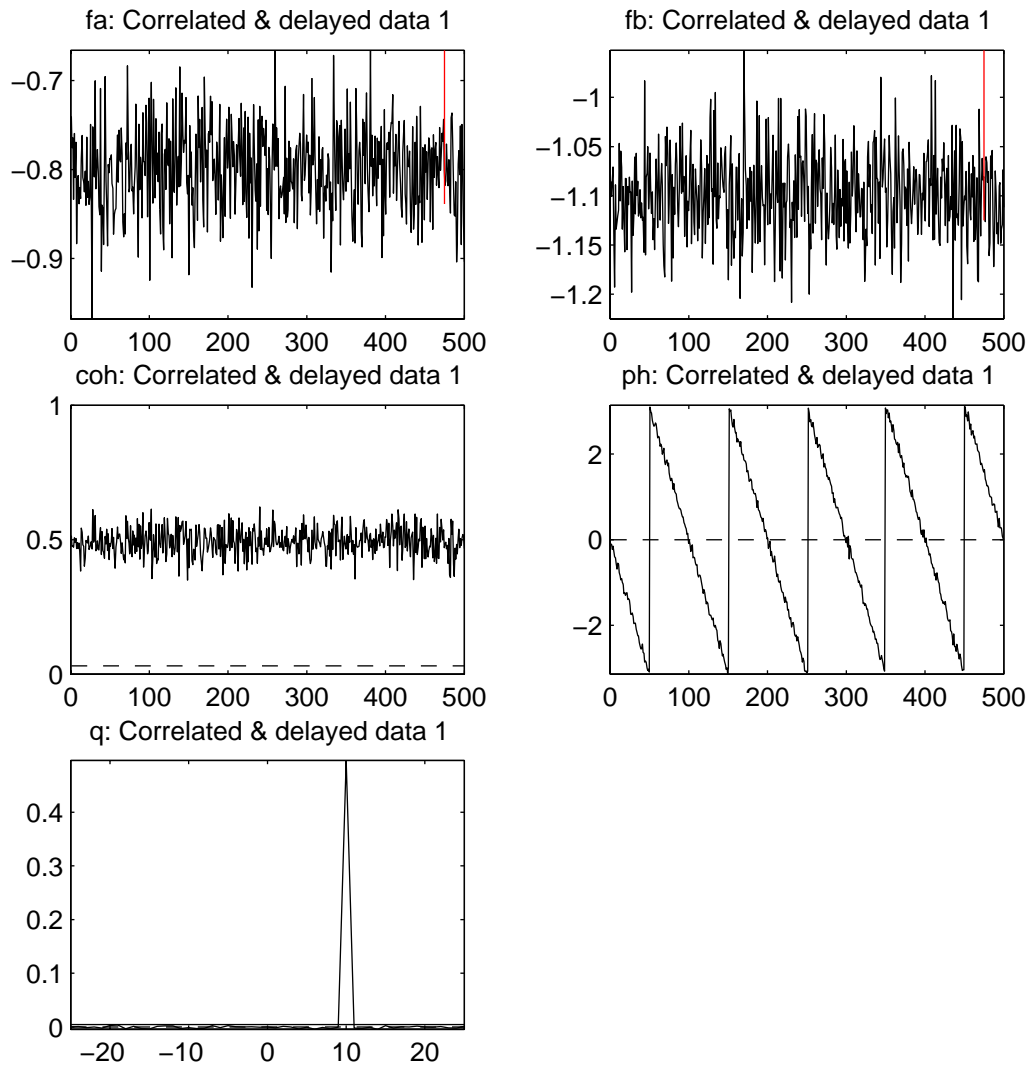


Figure 3. Output from `sp2a2_m`, for analysis of correlated and delayed signals. Layout as in Figure 1.

In figure 4, we look at the results of applying an 11 point moving average filter to the white noise data. The analysis reflects the characteristics of this filtering process. The input signal has a flat power spectrum as before, the output spectrum is now a monotonically decreasing function of frequency. The coherence tails off with frequency as the filter bandwidth is reached, and the input signal no longer influences the output signal. The phase is dominated by the delay in the moving average filter. The cumulant density gives an indication of the span of dependency of the filter in the time domain, in this case from 0 to 10 ms.

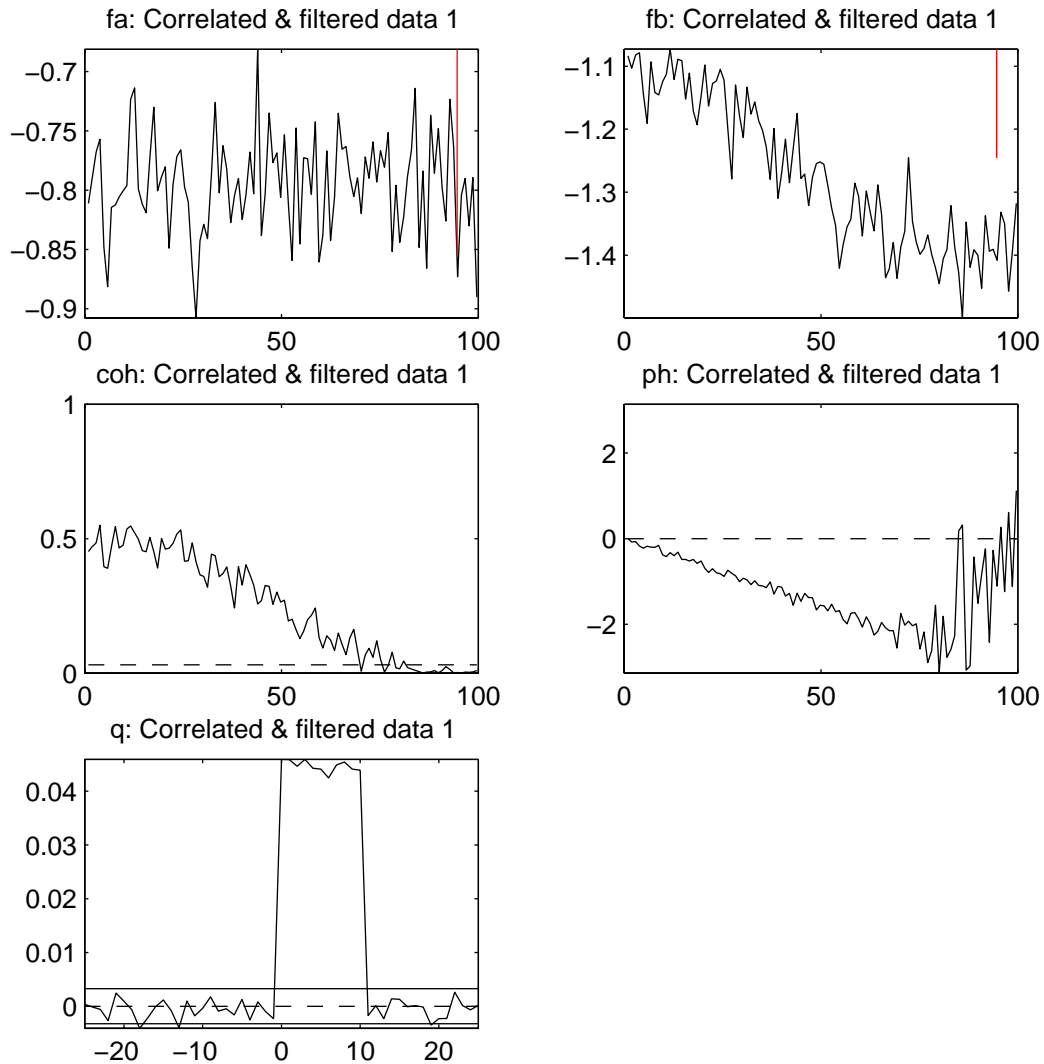


Figure 4. Output from `sp2a2_m`, for analysis of linearly filtered white noise. Layout as in Figure 1.

Figure 5, below, illustrates the results from analysis of two independent spike trains. Since they have been generated independently, these spike trains are independent and thus show no correlation. The second spike train has fewer harmonic components in its spectrum, this reflects the larger standard deviation of the inter-spike-intervals for this spike train compared with the first.

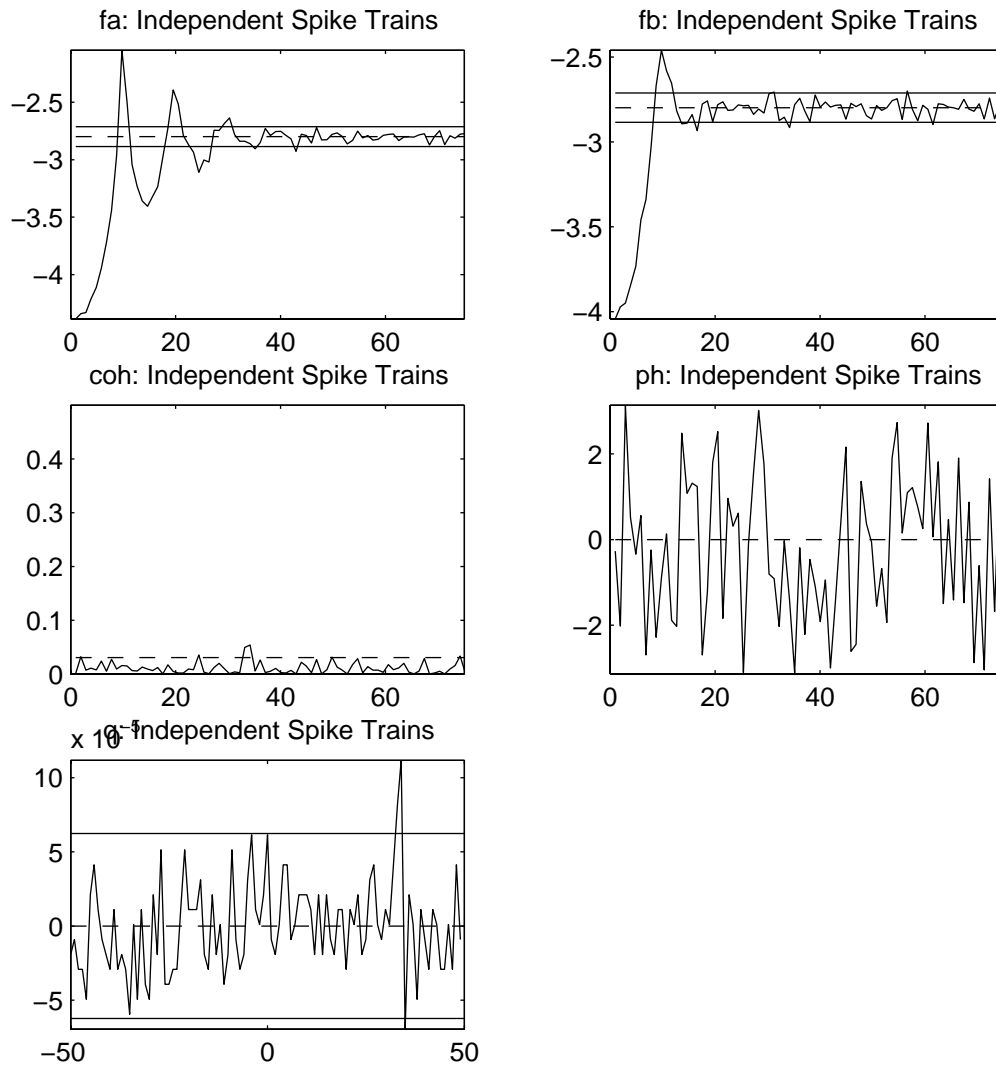


Figure 5. Output from `sp2_m`, for analysis of independent spike trains. Layout as in Figure 1.

Figure 6, below, illustrates the results from analysis of one spike train and one time series. Since the two sequences have been generated independently, they are independent and thus show no correlation. The spectrum of the spike train is shown at top left, the spectrum of the time series is shown at top right.

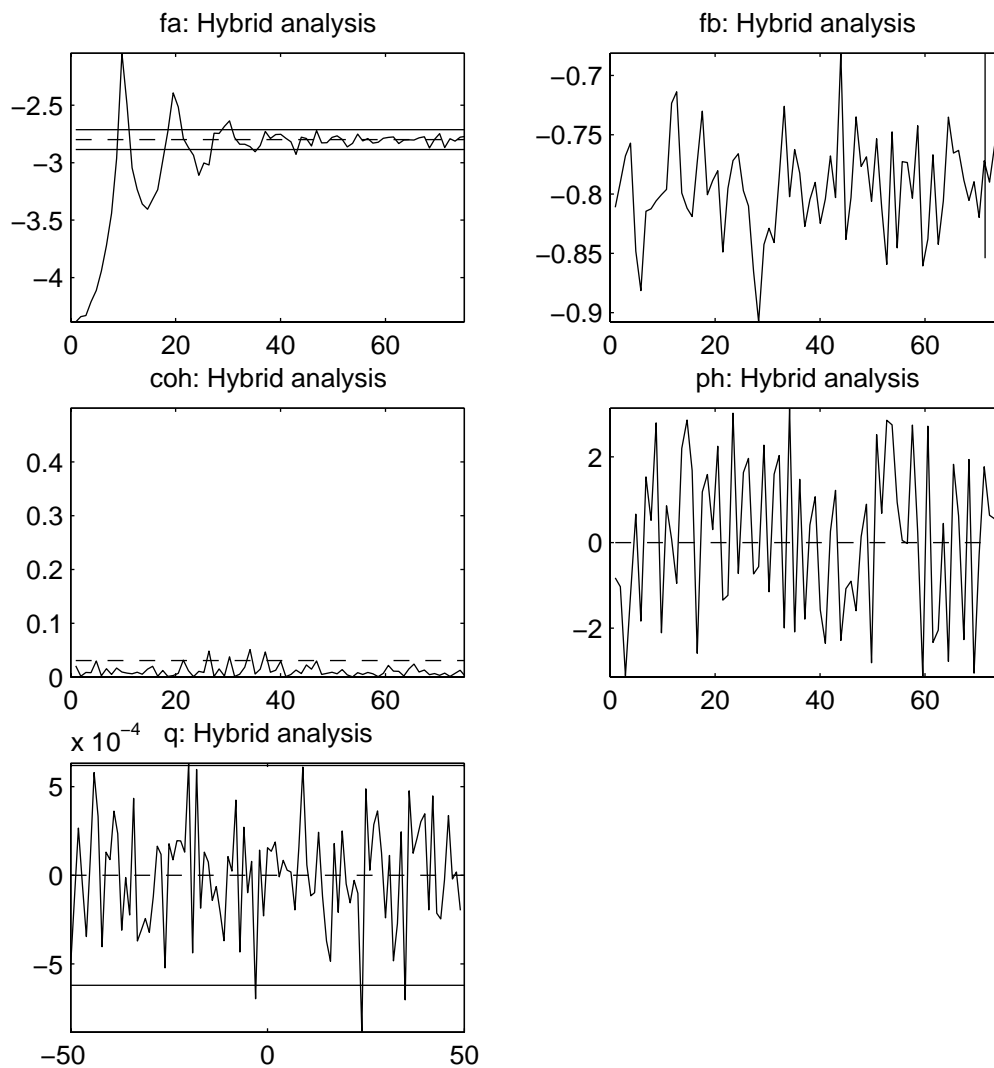


Figure 6. Output from `sp2a_m`, for analysis of one spike train and one time series. Layout as in Figure 1.

NeuroSpec 1.0. Guide, Copyright © 2002, David M. Halliday. Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved. This Version 1.0, 22/11/2002.

The URL of this document is: <http://www.neurospec.org/neurospec10.pdf>

The NeuroSpec home page is at: <http://www.neurospec.org/>

Contact: contact@neurospec.org