NeuroSpec 2.2 User Guide

David M. Halliday

Contents

1 Introduction

NeuroSpec 2.2 is a set of MATLAB functions and scripts to implement adaptive spectral tracking to estimate time varying coherence in segmented time series data. The approach is referred to as the *z-tracker*, it operates in the z-domain. The software implements the analysis framework described in [1]. It can also be applied to point process data. The software includes functions to calculate a locally smoothed estimate of coherence for each segment of data. Local smoothing is achieved using adaptive Kalman filtering of single segment coherence estimates. These single segment coherence estimates are obtained using multi-taper spectral analysis within each segment. Four categories of routine are included in the software:

1. Core analysis routine for estimating coherence over segments.
2. Plotting routines for time varying coherence: single frequency, average over frequencies, fixed segment or as a time-frequency heat map.
3. Supporting functions to generate surrogate data with known coherence structure, and calculate performance metrics.
4. Demonstration scripts using surrogate data with known coherence.

2 Software requirements

These routines require the MATLAB system to be installed. Version 7.x or higher is required to run NeuroSpec 2.2. NeuroSpec 2.2 routines have been tested on PC versions 8.1 and 9.2. No additional toolboxes are required, only the base system is needed. If you have an older version of MATLAB please read the FAQ page on the NeuroSpec web site.

3 Installation, quick start guide & getting help

To install the software:

1. Download and Unzip the file `neurospec22.zip`.
2. Install the directory `neurospec22` in your MATLAB path. See the MATLAB Help system for details of how to do this. In MATLAB 8.x the Set Path option is accessed from the HOME tab.
3. Change the MATLAB working directory to folder `neurospec22_demos` and run the demonstration scripts, starting with `ztrack_demo1`, see section 8.1

Once installed, additional help is available via the MATLAB help system. The command `helpwin neurospec22` should display a hyperlinked list of files, similar to that in Appendix A. Help for individual commands is available using the `help` or `helpwin` commands followed by the function name. The MATLAB source for the demonstration scripts in directory `neurospec22_demos` should provide additional guidance in the use of the software.

3.1 Relationship to earlier versions of NeuroSpec

All source files in NeuroSpec 2.2 are new files and can safely be copied into an existing installation of an earlier version of NeuroSpec. The one exception is the MAT file `NS_dpss_dat.mat` which is identical to the file included in NeuroSpec 2.1x. Only a single copy of this file is required, either copy can be used.

4 Format of raw data in MATLAB

This section describes the format that raw data should be input to the NeuroSpec routines. NeuroSpec can process either time series or point process (spike train) data. The formats specified below should readily accommodate most types of experimental and/or simulated data. We adopt the MATLAB convention of specifying multi-channel data in column format. *Note*: Failure to specify input data in this format may give incorrect results.

4.1 Time series data

Times series data in NeuroSpec 2.2 should be in the same format as in NeuroSpec 2.0 and NeuroSpec 2.11. Readers familiar with NeuroSpec 2.0 can skip to point-process data in section 4.2. Time series data is assumed to represent regularly sampled data with a known sampling rate. Integer or real values can be processed. Time series data should be in column format. Each column should represent one signal, with one sample per row. The sampling rate is input to the analysis - this, in conjunction with the FFT segment length, defines the spacing of the Fourier frequencies returned after analysis. Both single column and 2D matrices can be processed. Each column in a 2D matrix should contain data from one signal. Each row in a 2D matrix should contain data samples at the same time instant from several channels. Two single columns of data passed to the z-tracker analysis routine should have the same number of data points in each column and be from simultaneously sampled signals.

4.2 Point process (spike train) data

The handling of Point Process (Spike Train) data in NeuroSpec 2.2 is the same as in NeuroSpec 2.11, which differs from that in NeuroSpec 2.0. Spike train data should be in column format where spike timings are mapped into a 0/1 time series. This format represents spike trains as a regularly sampled sequence where the occurrence of a spike in a particular time bin is represented by the value 1, all other time bins have the value 0.

4.2.1 Conversion of point process data from NeuroSpec 2.0 format

The conversion of spike train data from that used for the routines in NeuroSpec 2.0 is straightforward, and can be illustrated by an example. If a vector of integer spike times is contained in the MATLAB variable: `sp1`, and assuming the spike train has duration: `sec_tot` (seconds), with spike times specified as an integer multiple of a sampling rate: `rate` (samples/sec), then the following MATLAB commands will generate a vector `dat_sp1` which can be used in the NeuroSpec 2.2 routines.

```
dat_sp1=zeros(sec_tot*rate,1);

dat_sp1(sp1)=1;
```

5 Using the **NeuroSpec** 2.2 core analysis routine

One core analysis function is provided to undertake bivariate z-tracker analysis:

`sp2a2_zt` Core function, uses `sp2_fn2_zta`, `sp2_fn2_ztb`, `zt_bias_v1` and `zt_var_v1`.

The z-tracker analysis implemented in function `sp2a2_zt` uses an approach similar to Type 0 analysis in NeuroSpec 2.0. The bivariate data is split into L non overlapping segments, each containing T data points. The total number of samples analysed is $R = LT$. Only complete segments are analysed, data points at the end of the record that do not make a complete segment are *not* included in the analysis. The segment length T is specified as a power of 2. The additional data segmentation strategies in NeuroSpec 2.0 (referred to as Type 1 and Type 2 analysis) are not supported in NeuroSpec 2.2. The function `sp2a2_zt` returns separate coherence estimates for each segment, with L coherence estimates combined in a single matrix. These time varying estimates are obtained by applying adaptive Kalman filtering or filtering and smoothing to multi-taper single segment coherence estimates as described in [1]

The use of the core routine can be illustrated by an example. Suppose you have in MATLAB the following variables:

`x, y` Two time series (or 0/1 spike train signals) of equal length.

Suppose the sampling rate for all signals is 1000/sec, in MATLAB: `rate=1000`. The segment length in NeuroSpec is specified as a power of 2, define the segment length for the FFT as 2^7 (128 points), in MATLAB: `seg_pwr=7`. For z-tracker coherence analysis between the signals x and y use the command:

```
[coh,cl]=sp2a2_zt(x,y,rate,seg_pwr);
```

The order in which the signals `x` and `y` are passed to `sp2a2_zt` is not important for coherence analysis, the results is the same if the order is swapped. In general this is not the case for bivariate NeuroSpec analysis, where order can impact phase and cross covariance estimates.

5.1 Using additional options

Options are controlled in the same way as earlier versions of NeuroSpec using an options string. Thus if you wish to rectify both channels and then perform a linear de-trend on both channels prior to analysis use the command:

```
[coh,cl]=sp2a2_zt(x,y,rate,seg_pwr,'t2 r2');
```

The options string can also be used to alter the values of α in the z-tracker, which controls the level of smoothing in the Kalman filtering, with larger values of alpha giving more smoothing. The value of α should be in the range $[0, 1]$. In NeuroSpec 2.2 the value is restricted to the range $0.1 \leq \alpha \leq 0.9$, the default value is $\alpha = 0.9$. A detailed analysis of the effects of changing α can be found in [1], which considers values in the range $0.1 \leq \alpha \leq 0.9$. The options string can also select whether to use only Kalman filtering (or tracking) as opposed to the default of Kalman filtering and smoothing, this uses the option `'T'`. To undertake a z-tracker analysis with $\alpha = 0.1$ and using only Kalman filtering use the option string `'a0.1 T'`:

```
[coh,cl]=sp2a2_zt(x,y,rate,seg_pwr,'a0.1 T');
```

Note that lower case `t` controls the linear de-trend option, in common with earlier versions of NeuroSpec whereas upper case `T` controls the Kalman filtering option. These can be combined in a single options string if required, e.g. `'t2 a0.1 T'`. The difference between Kalman filtering and Kalman filtering and smoothing is that the latter uses both forward and backward passes over the L segments of data, whereas filtering only uses a forward pass. Unless there are good reasons we recommend using the default filtering and smoothing. The differences between the two approaches are discussed in [1].

5.2 Using point process (spike train) data

Spike train data can be analysed using the routine `sp2a2_zt`. The format of the data is explained in section 4.2.1. Using this approach also supports analysis of hybrid (mixed time-series/point-process) data.

6 Outputs from the NeuroSpec 2.2 core routine

The function `sp2a2_zt` returns two variables. The outputs are:

1. `coh` A matrix of time varying coherence.
2. `cl` A structure containing confidence limits and other analysis parameters.

The coherence matrix has 1 column for each segment, containing the z-tracker coherence estimate for that segment. For segment size T and L segments the matrix is size $\left[\left(\frac{T}{2} - 1\right), L\right]$. These variables are not compatible with previous versions of NeuroSpec, so cannot be used with the plotting routines in previous versions (2.0x, 2.1x). For further details and the format of these variables see the source file `sp2a2_zt.m`.

7 Plotting

The z-tracker analysis returns estimates of coherence over segments in a 2-D matrix. Plotting functions are included which plot coherence at a fixed segment or plot the variation in coherence over segments. The latter can either be averaged over frequencies or for a single frequency. Four plotting functions are included:

1. `pspzt_ch1` Plots coherence over segments, single frequency or average over frequencies.

2. pspzt_ch1_target As No 1, includes target coherence for surrogate data.
3. pspzt_ch1_seg Plot of coherence for one specific segment over specified frequency range.
4. pspzt_ch1_tf Heat map of coherence over all segments and specified frequency range.

Assuming that [coh, cl] are returned from a z-tracker analysis, an example usage of function pspzt_ch1 would be:

```
pspzt_ch1(coh,cl,freq_ind,ch_max)
```

where freq_ind contains the index or index range to plot (as integers) and ch_max is the scaling to apply to the coherence axis ($0 \leq$ ch_max ≤ 1). To plot a single frequency use freq_ind $= 1$ to plot the first frequency, freq_ind $= 2$ for the second, etc. To plot a range of frequencies use a vector of values, for example for $T = 128$ use freq_ind $= (1 : 63)$ to plot the average coherence over all possible, $(T/2 - 1)$, frequencies. Note that frequencies are specified as indices rather than specific values in Hz. The specific values for each frequency are contained in the field ost_freqs of the structure cl. to see a list of frequencies (in Hz) in the command window type: cl.ost_freqs. The demonstration script ztrack_demo1 illustrates usage of all 4 plotting functions.

8 Demonstration scripts

Three demonstration scripts are included as part of NeuroSpec 2.2. The first two, ztrack_demo1 and ztrack_demo2 are demonstrations on single sets of surrogate data. The third, ztrack_ramp_demo1 undertakes an extensive simulation study with 100 repetitions across 4 values of segment length T and 8 values of α controlling the Kalman filter smoothing. This duplicates the simulation results presented in [1]. Also included is a script ztrack_plot_var_bias which plots the lookup tables used for variance and bias, shown in figure 1 in [1]. All demonstration scripts are in the directory neurospec22_demos.

8.1 Script ztrack_demo1

This demonstrates z-tracker analysis on 2 channel Gaussian data containing a linear increase-decrease pattern of coherence, the modulation of coherence is constant across all frequencies. The script generates five figures. The first is the target coherence illustrating the linear increase-decrease over time, using function ztrack_dat_ramp1. The second figure shows the estimated coherence over segments averaged across all frequencies and for a single frequency, as illustrated in figure 1 for a sample run. These plots also include point-wise confidence limits derived from the Kalman filter error estimate. The third figure is the same as the second with the addition of the (known) target coherence.

The fourth figure generated by the script shows coherence estimates at specific segments across the record, chosen when the target coherence is close to 1, around 0.5 and close to zero, respectively. An example of this is shown in figure 2. The last figure shows plots of the data with the time-varying coherence over segments plotted as a heatmap. An example of this is shown in figure 3.

8.2 Script ztrack_demo2

This script is based on ztrack_demo1, using the same type of surrogate data with a linear-increase decrease of target coherence in Gaussian data. It demonstrates the effects of changing the level of smoothing in the Kalman filter, controlled by the parameter α and the effects of using tracking (forward pass only) and smoothing (forward and reverse passes across data) in the Kalman filter. Four values of α are used: $[0.1,\ \exp(-1),\ \exp(-0.5),\ 0.9]$. These illustrate the full range of values for α that can be incorporated in the analysis. The minimum value $\alpha = 0.1$ applies the least smoothing in the Kalman filter, $\alpha = 0.9$ represents maximum smoothing. The intermediate values correspond to time constants of 1 and 2 segments in the exponential smoothing of the process noise q_l applied to the Kalman filter, see equation (7) in [1] for more details. Figure 4 shows results from a sample run with the estimated z-tracker coherence averaged across frequencies for these 4 values of α, figure 5 shows sample single frequency z-tracker estimates.

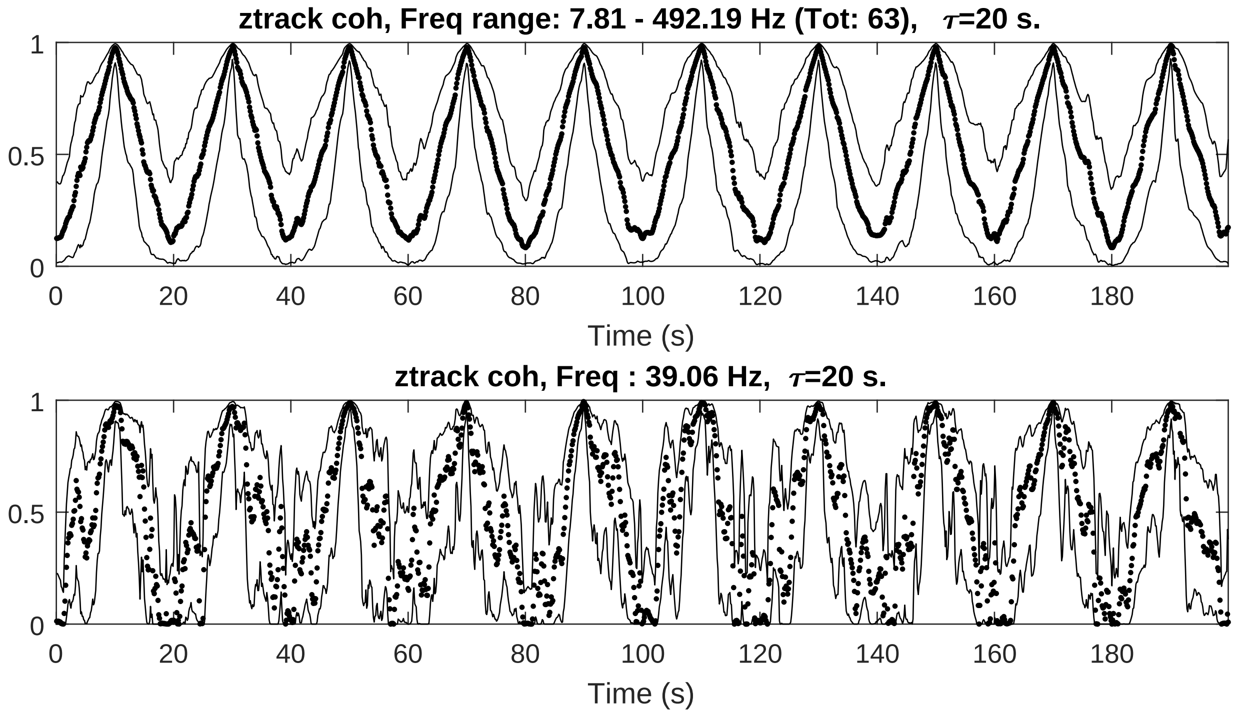

Figure 1: Plots generated by `pspzt_ch1` of output from `sp2a2_zt` in demonstration script `ztrack_demo1` for Gaussian data with a linear increase-decrease pattern of coherence over 20 second time scale ($\tau = 20$s). Top graph shows the z-tracker coherence estimate averaged across all $(T/2 - 1)$ frequencies, 63 in this case. The lower graphs shows the coherence estimate for a single frequency, index 5, corresponding to 39Hz with a segment length $T = 128$ and an assumed sampling rate of 1000/sec. The estimated coherence is indicated by a dot for each segment . The solid horizontal lines indicate the upper and lower 95% confidence limits based on the Kalman filter error estimate for each segment.

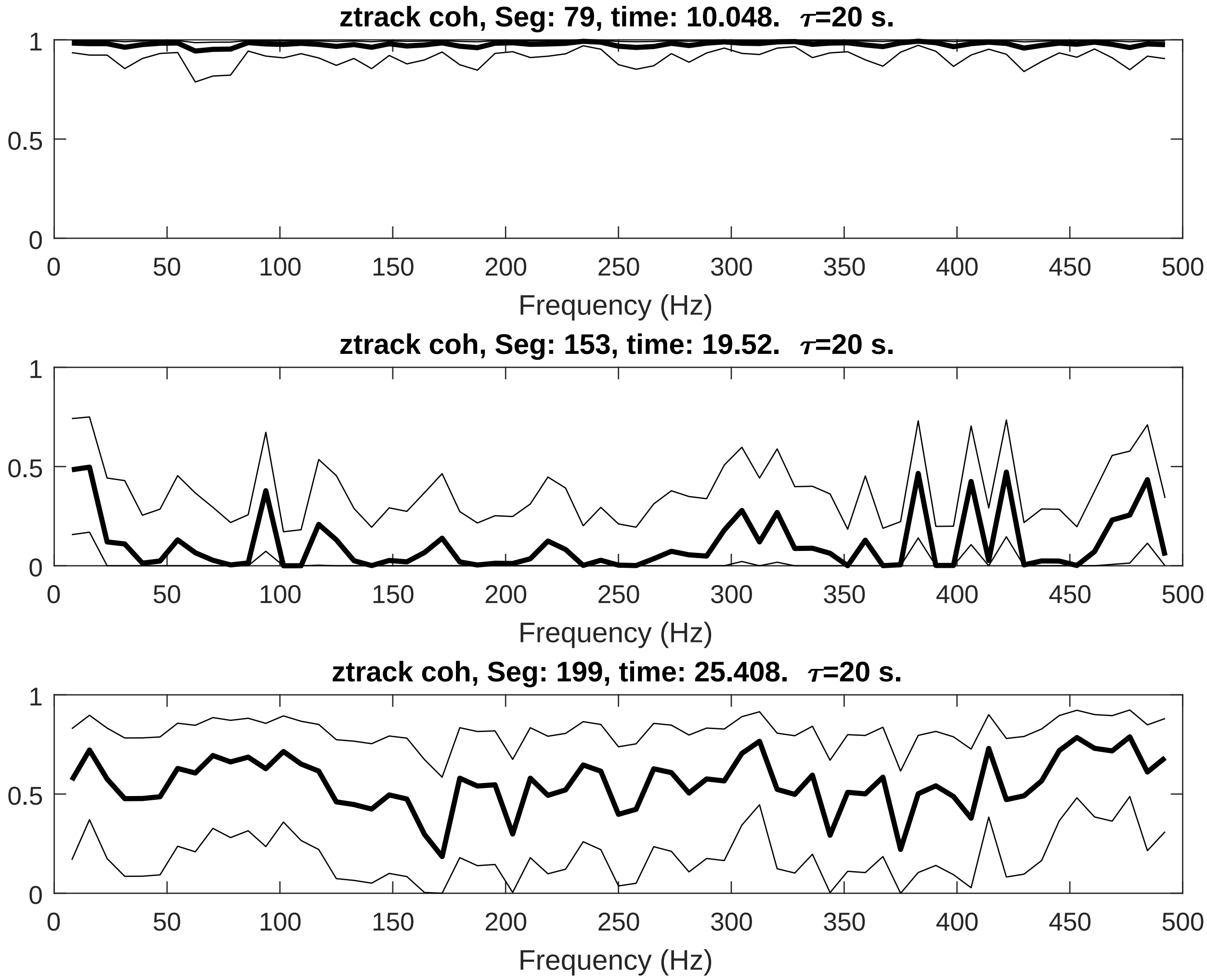

Figure 2: Plots generated by `pspzt_ch1_seg` of output from `sp2a2_zt` in demonstration script `ztrack_demo1` for Gaussian data with a linear increase-decrease pattern of coherence. The three plots show coherence at fixed segments with varying levels of target coherence, which is constant across all frequencies. The estimated coherence for each segment is indicated by a thick line. The thin lines indicate the upper and lower 95% confidence limits based on the Kalman filter error estimate for each segment.

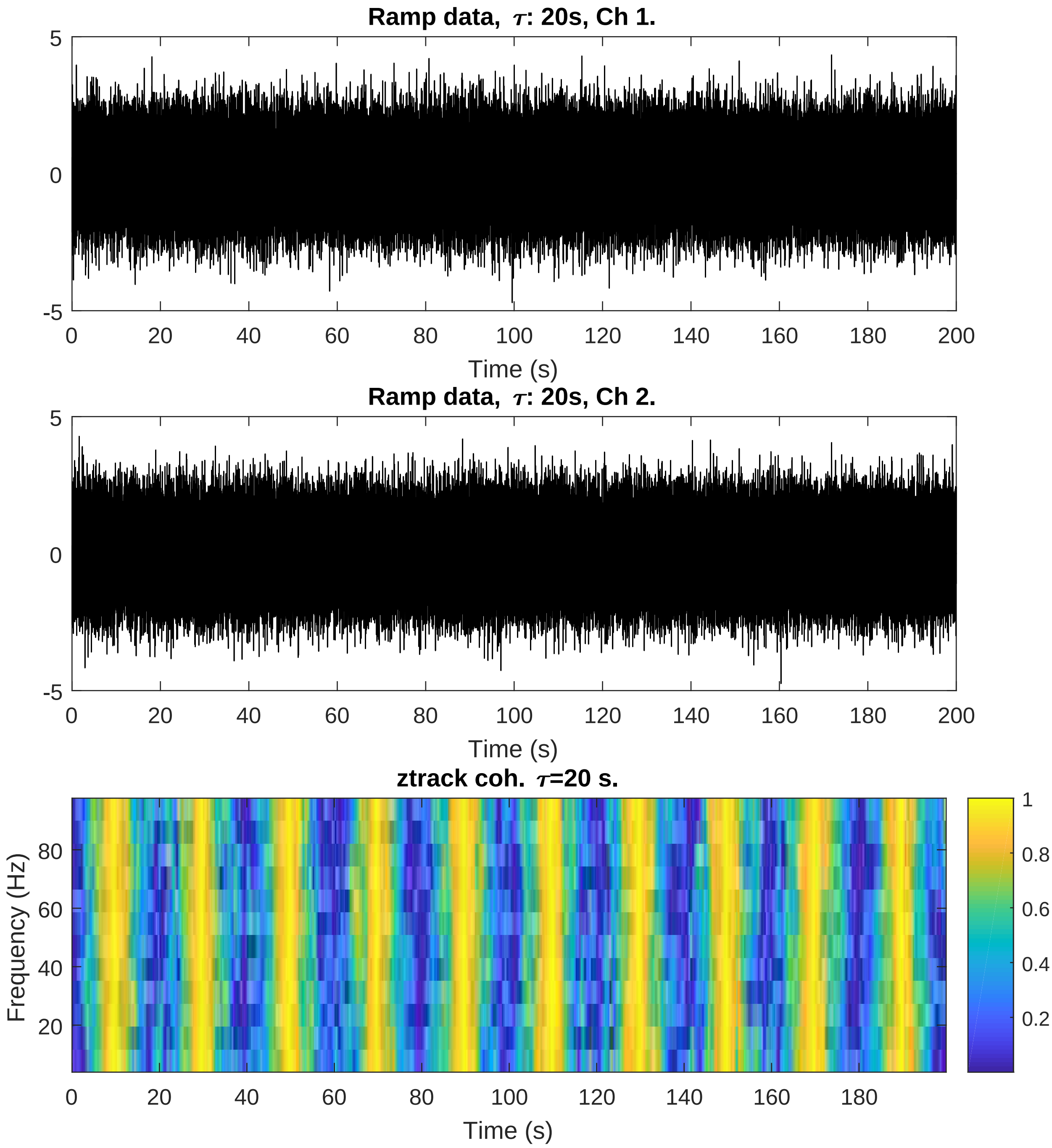

Figure 3: Plots of raw data (upper, middle) and plot generated by `pspzt_ch1_tf` (lower) of output from `sp2a2_zt` in demonstration script `ztrack_demo1` for Gaussian data with a linear increase-decrease pattern of coherence. The z-tracker coherence estimate is shown as a heat-map in the time-frequency plane in the lower plot. The magnitude of the coherence is indicated by the scale bar on the right.

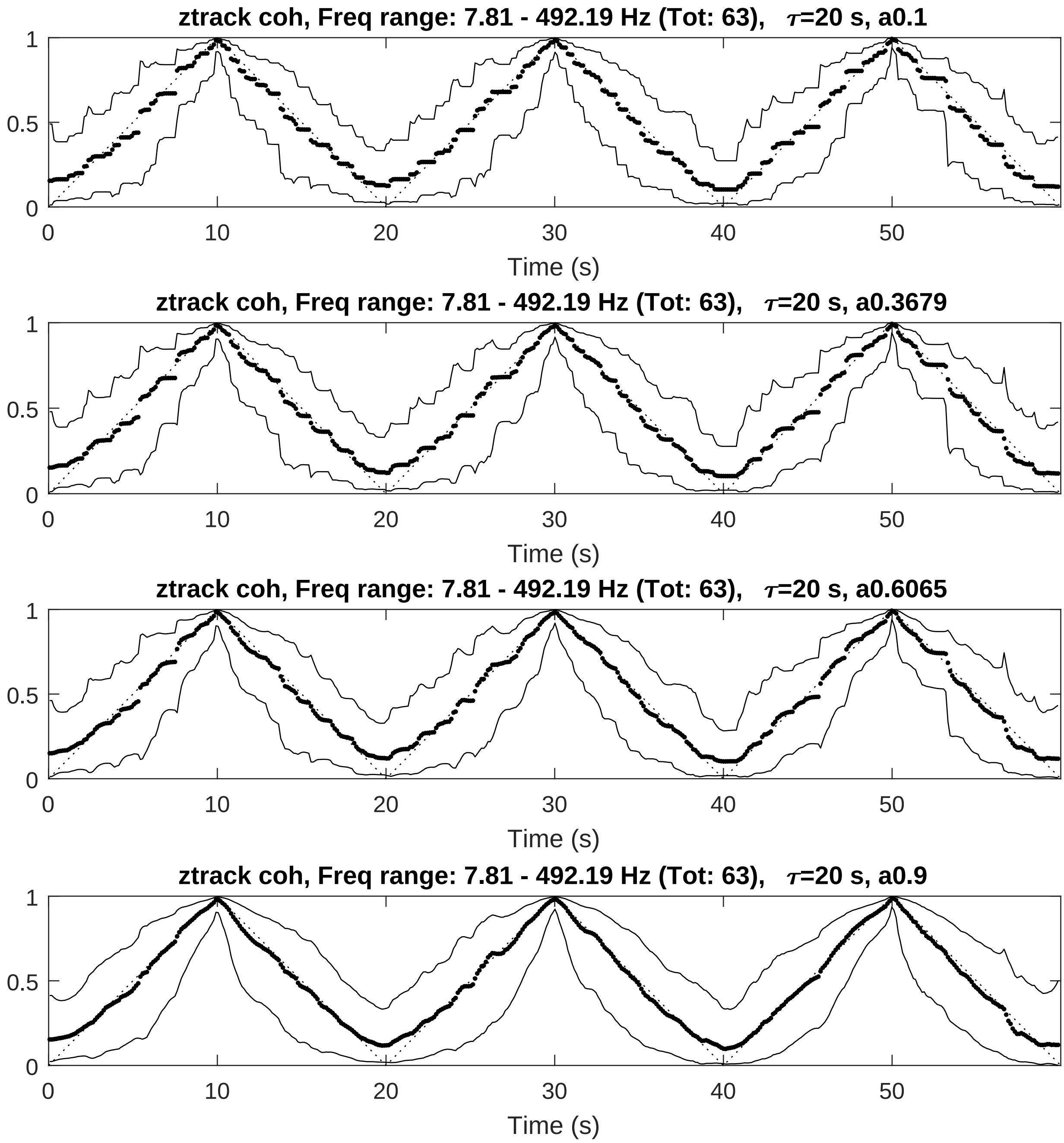

Figure 4: Plots generated by `pspzt_ch1` of output from `sp2a2_zt` in demonstration script `ztrack_demo2` for Gaussian data with a linear increase-decrease pattern of coherence over 20 second time scale ($\tau = 20$s). The graphs show the z-tracker coherence estimate averaged across all $(T/2-1)$ frequencies, 63 in this case for different values of α: $[0.1,\ \exp(-1),\ \exp(-0.5),\ 0.9]$. The estimated coherence for each segment is indicated by a thick dot. The solid horizontal lines indicate the upper and lower 95% confidence limits based on the Kalman filter error estimate for each segment. The target coherence is indicated by a thin dotted line.

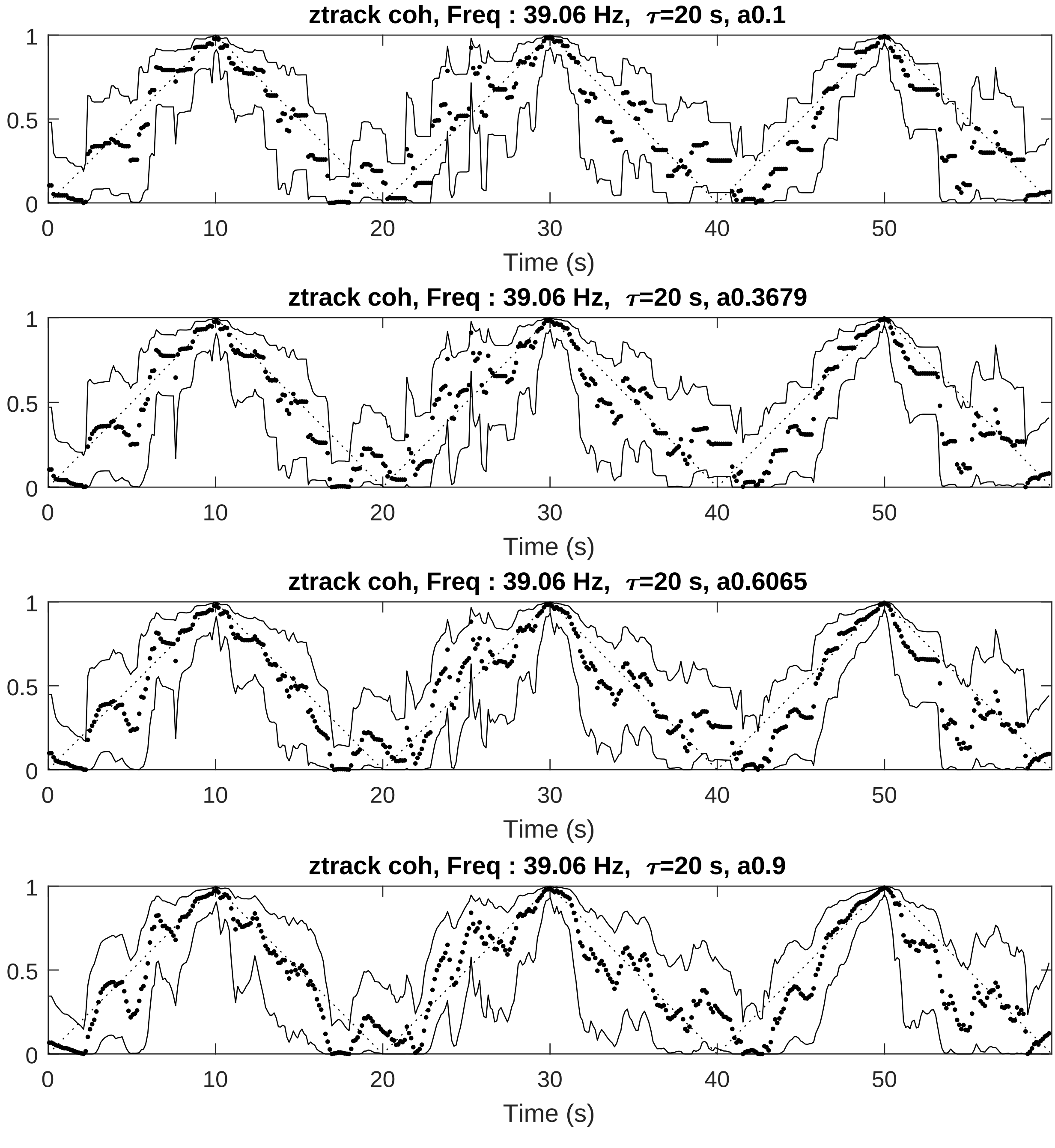

Figure 5: Plots generated by `pspzt_ch1` of output from `sp2a2_zt` in demonstration script `ztrack_demo2` for Gaussian data with a linear increase-decrease pattern of coherence over 20 second time scale ($\tau = 20$s). The graphs show the z-tracker coherence estimate for a single frequency, index 5 corresponding to 39 Hz with $T = 128$ and an assumed sampling rate of 1000/s, for different values of α: [0.1, $\exp(-1)$, $\exp(-0.5)$, 0.9]. The estimated coherence for each segment is indicated by a thick dot. The solid horizontal lines indicate the upper and lower 95% confidence limits based on the Kalman filter error estimate for each segment. The target coherence is indicated by a thin dotted line.

8.3 Script `ztrack_ramp_demo1`

This script creates some of the surrogate data that was presented in [1]. As supplied it will take 10-20 minutes to run and generates 2.8GB of data in MATLAB. It uses the same type of surrogate data as `ztrack_demo1`, with constant coherence across all frequencies, linearly modulated over time to cover the specified range, [0, 1] in this case. The script uses four values for segment length T: [128, 256, 512, 1024], four values of α: $[0.1,\ \exp(-1),\ \exp(-0.5),\ 0.9]$ and uses both tracking and smoothing in the Kalman filter. Each configuration is run 100 times. Two additional scripts are included to plot the data generated by `ztrack_ramp_demo1`:

1. `ztrack_ramp_demo1_plot1` Generates 16 figures.
2. `ztrack_ramp_demo1_plot_msd1` Generates 5 figures.

Each figure in `ztrack_ramp_demo1_plot1` has 10 panels, each panel shows the z-tracker estimate for one individual trial. The first 8 figures show z-tracker estimates averaged across 31 frequency values with $T = 128$. The second 8 figures show z-tracker estimates for a single frequency. These figures are similar to figs 2 and 3 in [1]. The 5 figures generated by `ztrack_ramp_demo1_plot_msd1` are:

1. MSD box plots for varying T, figure 4 in [1].
2. MSD box plots for varying α, similar to figure 5 in [1].
3. MSD box plots for single frequency values in z-tracker, figure 6 in [1].
4. Box plots of median P_l for z-tracker, similar to figure 13 in [1].
5. Histogram of P_l over segments for single trial, figure 14 in [1].

8.3.1 Modifications to `ztrack_ramp_demo1`

As supplied the script `ztrack_ramp_demo1` is configured to generate results similar to those presented in figures 2 - 6 in [1]. Figures 7 - 10 in [1] explore the behaviour of the z-tracker when the coherence is modulated over a shorter time scale, with target coherence modulated from 0 to 1 over 1 second. This can be achieved using `ztrack_ramp_demo1` by altering the variables `tau` and `n_rep`, see lines 57, 58 of the script. Figures 11 - 12 in [1] use surrogate data where the target coherence is consists of section with a linear increase and step decrease, using randomly determined transition points. The script `ztrack_ramp_demo2` generates data with this type of target coherence, which can be analysed using the same approach as `ztrack_dat_ramp1`.

9 Relationship to NeuroSpec 2.01 and NeuroSpec 2.11

The analysis in NeuroSpec 2.2 is not compatible with any of the plotting routines in NeuroSpec 2.01, 2.11. Similarly the plotting routines in NeuroSpec 2.2 are not compatible with any of the analyses routines in NeuroSpec 2.01, 2.11.

An initial exploration of any data set assuming stationarity can be undertaken using the routines in NeuroSpec 2.11 with average periodogram or multi-taper analysis. The handling of point-process (spike train) data in NeuroSpec 2.2 is similar to that in NeuroSpec 2.11, see section 4.2.1.

10 Licensing

NeuroSpec is free software. It is covered by the GNU General Public Licence, a copy of which is included in the distribution file `neurospec22.zip`.

11 Further information

This user guide focuses on the practical aspects of using the NeuroSpec software. The references below can be consulted for further details of the analytical framework. The z-tracker is described in detail in [1]. Two tutorial reviews [2, 3] provide background for bivariate and multivariate analysis using average periodograms. Details of how to download copies of these articles can be found on the NeuroSpec web site www.neurospec.org. Please refer to the articles below (as appropriate) in any scholarly publications resulting from the use of NeuroSpec software.

References

[1] Halliday, D. M., Brittain, J.-S., Stevenson, C. W., & Mason, R. (2018). Adaptive spectral tracking for coherence estimation: the z -tracker. Journal of Neural Engineering, 15(2), 26004. 10.1088/1741-2552/aaa3b4.

[2] Rosenberg, J.R., Amjad, A.M., Breeze, P., Brillinger, D.R., & Halliday, D.M. (1989). The Fourier approach to the identification of functional coupling between neuronal spike trains. *Progress in Biophysics and molecular Biology* **53**, 1-31.

[3] Halliday, D.M., Rosenberg, J.R., Amjad, A.M., Breeze, P., Conway, B.A. & Farmer, S.F. (1995). 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.

12 Acknowledgements

NeuroSpec version 2.2 has been written by David Halliday. Thanks are due to the many people who have contributed to the development of the NeuroSpec framework over many years: Jay Rosenberg, Bernie Conway, Abdul Majeed Amjad, Alex Rigas, David Murray-Smith, Joe Lau, Peter Breeze, Simon Farmer, Jens Nielsen, Yang Zhan, John-Stuart Brittain, Carl Stevenson, Rob Mason. The work has been supported by grants from the Wellcome Trust, BBSRC, EPSRC & MRC.

The URL of this document is: `http://www.neurospec.org/neurospec22.pdf`
The NeuroSpec home page is at: `http://www.neurospec.org/`

Contact: david.halliday@york.ac.uk

A Appendix A: List of files

This Appendix gives an overview of the .m and .mat files included in NeuroSpec 2.2.

Core routines for z-tracker analysis	
sp2a2_zt.m	z-tracker analysis function.
sp2_fn2_zta.m	Called by sp2a2_zt, estimates single segment coherence.
sp2_fn2_ztb.m	Called by sp2a2_zt, z-tracker analysis on z values.
zt_var_v1.m	Look up table for single segment variance, used in sp2_fn2_ztb.
zt_bias_v1.m	Look up table for single segment bias, used in sp2_fn2_ztb.
Plotting functions	
pspzt_ch1.m	Plot of z-tracker coherence estimate over segments, single frequency or averaged over frequencies.
pspzt_ch1_target.m	As pspzt_ch1.m, includes target coherence.
pspzt_ch1_seg.m	Plot of z-tracker coherence estimate at fixed segment.
pspzt_ch1_tf.m	Plot of z-tracker coherence estimate as time-frequency heat map.
Supporting functions	
ztrack_dat_ramp1.m	Generates surrogate data with linear increase-decrease in target coherence.
ztrack_dat_ramp_step1.m	Generates surrogate data with linear increase and step decrease in target coherence.
zt_msd1.m	Calculates MSD between target and estimated coherence.
Demonstration scripts	
ztrack_plot_var_bias.m	Plots lookup tables for variance and bias, figure 1 in [1].
ztrack_demo1.m	Basic demonstration on surrogate data with linear increase-decrease in target coherence.
ztrack_demo2.m	Similar to ztrack_demo1, uses different levels of smoothing in Kalman filter.
ztrack_ramp_demo1.m	Extensive set of repeat runs on surrogate data.
ztrack_ramp_demo1_plot1.m	Plotting of z-tracker estimates for data generated by ztrack_ramp_demo1.
ztrack_ramp_demo1_plot_msd1.m	Plotting of MSD box plots for data generated by ztrack_ramp_demo1.
ztrack_ramp_demo2.m	Illustrates use of function ztrack_dat_ramp_step1.
MAT files	
NS_dpss_dat.mat	Data windows for multi-taper analysis, used in sp2_fn2_zta. *Note:* This file is identical to that in NeuroSpec 2.11.