**Documentation for function *tfa\_car.m***

This function is a Matlab implementation of the TFA as specified in the White Paper (2015). An example of use on the supplied data (*tfa\_sample\_data.txt*) is given in *tfa\_demo.m* and results are reported at the end of this report.

**function tfa\_out=tfa\_car(ABP,CBFV,fs,params)**

Input parameters:

*ABP*: Arterial blood pressure (assumed to be in mmHg)

*CBFV*: Cerebral blood flow velocity (assumed to be in cm/s)

*fs*: Sampling frequency (assumed to be in Hz)

*params: A series of parameters that control TFA analysis (window-length, frequency bands …). If this is not provided, default values, corresponding to those recommended in the white paper, will be used. These default values are given below for each parameter.*

*params.vlf=[0.02,0.07]:* Limits of very low frequency band (in Hz). Note that to avoid double counting any frequency, values exactly at the upper limit will be excluded, but values exactly at the lower limit will be included.

*params.lf=[0.07,0.2]:* Limits of low frequency band (in Hz). Note that to avoid double counting any frequency, values exactly at the upper limit will be excluded, but values exactly at the lower limit will be included.

*params.hf=[0.2,0.5]:* Limits of high frequency band (in Hz). Note that to avoid double counting any frequency, values exactly at the upper limit will be excluded, but values exactly at the lower limit will be included.

*params.detrend=0:* linear detrending of data prior toTFA analysis (detrending is carried out as one continuous trend over the whole length of the recording, not segment-by-segment). 0: no, 1: yes. Note that the mean of the whole segment is removed for both ABP and CBFV, regardless of this setting.

*params.spectral\_smoothing=3:* The length, in samples, of the triangular spectral smoothing function.Note that this must be an odd number, to ensure that smoothing is symmetrical around the centre frequency*.*

*params.coherence2\_thresholds=[3:15;0.51,0.40,0.34,0.29,0.25,0.22,0.20,0.18,0.17,0.15,0.14,0.13,0.12]':*  The critical values (alpha=5%, bottom row of the matrix) for coherence for a number of windows (given by the top row of the matrix, here from 3 to 15). These values were obtained by Monte Carlo simulation, using the default parameter settings for the TFA analysis (Hanning window, overlap of 50% and 3-point spectral smoothing was assumed). These values should be recalculated for different settings. Note that if *params.overlap\_adjust=1,* the overlap will vary depending on the length of data. With an overlap of 60% (see below), the critical values increase by between 0.04 (for 3 windows) and 0.02 (for 15 windows).

*params.apply\_coherence2\_threshold=1:* Apply the thresholds given above to the TFA estimates. 0: no, 1: yes. All frequencies with magnitude-squared coherence below the threshold value are excluded from averaging when calculating the mean values of gain and phase across the bands. Note that low values of coherence are not excluded in the average of coherence across the bands.

*params.remove\_negative\_phase=1:* Remove (ignore) negative values of phase in averaging across bands. 0: no, 1: yes. Negative phase values are removed only for frequencies below the frequency given below, when calculating the average phase in bands.

*params.remove\_negative\_phase\_f\_cutoff=0.1:* The cut-off frequency below-which negative phase values are neglected (if the parameter above is set to 1).

*params.normalize\_ABP=0:* Normalize ABP by dividing by the mean and multiplying by 100, to *express ABP change in %.* Note that mean-values are always removed from ABP prior to analysis.

*params.normalize\_CBFV=0:* Normalize CBFV by dividing by the mean and multiplying by 100, to *express CBFV change in %.* Note that the band-average values of gain are always calculated both with and without normalization of CBFV, in accordance with the recommendations (see *tfa\_out.Gain\_vlf\_not\_norm*, *tfa\_out.Gain\_vlf\_norm, etc.)* Note also that mean-values are always removed from CBFV prior to analysis.

*params.window\_type='hanning':*  Chose window ‘hanning’ or ‘boxcar’.

*params.window\_length=102.4:* Length of the data-window, in seconds.

*params.overlap=59.99:* Overlap of the windows, in %. If *params.overlap\_adjust=1 (*see below), then this value may be automatically reduced, to ensure that windows cover the full length of data. Here 59.99% rather than 60% was chosen, so that with data corresponding to 5 windows of 100 s at an overlap of 50%, 5 windows are indeed chosen.

*params.overlap\_adjust=1:* Ensure that the full length of data is used (i.e. the last window finishes as near as possible to the end of the recording), by adjusting the overlap up to a maximum value given by *params.overlap.*  0: no, 1: yes.

*params.plot=1:* Plot graphs. 0: no, 1: yes.

*params.plot\_f\_range=[0,0.5]:* Range of frequencies to show in the plots.

*params.plot\_title='':* Title of plots.

Output values:

*tfa\_out* provides the values of all expected measures from TFA analysis

*tfa\_out.Mean\_abp* : Mean ABP

*tfa\_out.Std\_abp*: Standard deviation of ABP

*tfa\_out.Mean\_cbfv*: Mean CBFV

*tfa\_out.Std\_cbfv*: Standard deviation of CBFV

*tfa\_out.overlap*: overlap of windows (in %). Note that this may not be the precise value given in params (or the default value set), when overlap is changed slightly to avoid discarding data (see *params.overlap\_adjust*)

*tfa\_out. H*: The complex frequency response (transfer function) over the full frequency range

*tfa\_out.C*: The coherence (complex, not magnitude squared)

*tfa\_out.f*: Frequencies corresponding to each of the values of the frequency response (*tfa\_out.H)*, coherence (*tfa\_out.C*), and spectra (see next lines)

*tfa\_out.Pxx*: Powerspectrum of input signals (ABP)

*tfa\_out.Pyy:* Powerspectrum of output signal (CBFV)

*tfa\_out.Pxy:* Cross-powerspectral density of input and output

*tfa\_out.No\_windows*: Number of Windows used

*tfa\_out.Gain\_vlf*: average gain in very low frequency band (normalized or not, depending on the *params.normalize … setting)*

*tfa\_out.Phase\_vlf:* average phase (in degrees) in very low frequency band

*tfa\_out.Coh2\_vlf*: average magnitude-squared coherence in very low frequency band

*tfa\_out. P\_abp\_vlf*: ABP power in very low frequency band

*tfa\_out.P\_cbfv\_vlf*: CBFV power in very low frequency band

*tfa\_out.Gain\_lf*: average gain in low frequency band (normalized or not, depending on the *params.normalize … setting)*

*tfa\_out.Phase\_lf*: average phase (in degrees) in low frequency band

*tfa\_out.Coh2\_lf:* average magnitude-squared coherence in low frequency band

*tfa\_out.P\_abp\_lf:* ABP power in very low frequency band

*tfa\_out.P\_cbfv\_lf*: CBFV power in very low frequency band

*tfa\_out.Gain\_hf:* average gain in high frequency band (normalized or not, depending on the *params.normalize … setting)*

*tfa\_out.Phase\_hf*: average phase (in degrees) in high frequency band

*tfa\_out.Coh2\_hf*: average magnitude-squared coherence in high frequency band

*tfa\_out.P\_abp\_hf*: ABP power in high frequency band

*tfa\_out.P\_cbfv\_hf*: CBFV power in high frequency band

*tfa\_out.Gain\_vlf\_not\_norm*: average gain in very low frequency band (without CBFV normalization, i.e. usually cm.s-1.mmHg-1)

*tfa\_out.Gain\_lf\_not\_norm*: average gain in low frequency band (without CBFV normalization, i.e. usually cm.s-1.mmHg-1)

*tfa\_out.Gain\_hf\_not\_norm*: average gain in high frequency band (without CBFV normalization, i.e. usually cm.s-1.mmHg-1)

*tfa\_out.Gain\_vlf\_norm*: average gain in very low frequency band (with CBFV normalization, i.e. usually %.mmHg-1)

*tfa\_out.Gain\_lf\_norm*: average gain in low frequency band (with CBFV normalization, i.e. usually %.mmHg-1)

*tfa\_out.Gain\_hf\_norm*: average gain in high frequency band (with CBFV normalization, i.e. usually %.mmHg-1)

Results from using tfa\_car.m on the sample dataset: tfa\_sample.txt. The script *tfa\_demo.m* shows how to use the function and results are given below, including the output plots.

Left channel:





The red dotted lines show the mean values calculated over the respective frequency ranges. The black dashed line shows the critical value of coherence for the 5 windows used here. Note that mean gain and phase is only calculated where coherence is above the dashed line.

The output values from the function:

tfa\_out =

Mean\_abp: 70.0036

Std\_abp: 4.3092

Mean\_cbfv: 64.9327

Std\_cbfv: 2.9676

overlap: 50

H: [1024x1 double]

C: [1024x1 double]

f: [1024x1 double]

Pxx: [1024x1 double]

Pyy: [1024x1 double]

Pxy: [1024x1 double]

No\_windows: 5

Gain\_vlf: 0.6760

Phase\_vlf: 52.9658

Coh2\_vlf: 0.5054

P\_abp\_vlf: 6.2455

P\_cbfv\_vlf: 3.2171

Gain\_lf: 0.9579

Phase\_lf: 25.4391

Coh2\_lf: 0.6171

P\_abp\_lf: 1.5583

P\_cbfv\_lf: 2.2532

Gain\_hf: 1.1988

Phase\_hf: 9.3763

Coh2\_hf: 0.5730

P\_abp\_hf: 0.2131

P\_cbfv\_hf: 0.3039

Gain\_vlf\_not\_norm: 0.6760

Gain\_lf\_not\_norm: 0.9579

Gain\_hf\_not\_norm: 1.1988

Gain\_vlf\_norm: 1.0410

Gain\_lf\_norm: 1.4752

Gain\_hf\_norm: 1.8462

Right CBFV



tfa\_out =

Mean\_abp: 70.0036

Std\_abp: 4.3092

Mean\_cbfv: 61.5967

Std\_cbfv: 2.7737

overlap: 50

H: [1024x1 double]

C: [1024x1 double]

f: [1024x1 double]

Pxx: [1024x1 double]

Pyy: [1024x1 double]

Pxy: [1024x1 double]

No\_windows: 5

Gain\_vlf: 0.5115

Phase\_vlf: 35.6390

Coh2\_vlf: 0.4941

P\_abp\_vlf: 6.2455

P\_cbfv\_vlf: 2.6237

Gain\_lf: 0.8792

Phase\_lf: 31.8909

Coh2\_lf: 0.4565

P\_abp\_lf: 1.5583

P\_cbfv\_lf: 1.9993

Gain\_hf: 1.1033

Phase\_hf: 3.0658

Coh2\_hf: 0.4774

P\_abp\_hf: 0.2131

P\_cbfv\_hf: 0.3292

Gain\_vlf\_not\_norm: 0.5115

Gain\_lf\_not\_norm: 0.8792

Gain\_hf\_not\_norm: 1.1033

Gain\_vlf\_norm: 0.8304

Gain\_lf\_norm: 1.4273

Gain\_hf\_norm: 1.7911

David Simpson, September 2015

Ds@isvr.soton.ac.uk