SPIKY_loop --- Copyright Thomas Kreuz, Nebojsa Bozanic; February 2015

'SPIKY_loop' is complementary to the graphical user interface 'SPIKY'. Both programs can be used to calculate time-resolved measures of spike train (dis)similarity (ISI- and SPIKE-distance as well as SPIKE synchronization) between two (or more) spike trains. However, whereas SPIKY was mainly designed to facilitate the detailed analysis of one dataset, 'SPIKY_loop' is meant to be used in order to compare the results for many different datasets (typically in some kind of loop).

'SPIKY_loop' is the main program where the variables are set and from where the main funtion 'SPIKY_loop_f_distances' is called.

More information on the program and the spike train distances can be found under

"http://www.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/SPIKY.html" and/or in

Kreuz T, Mulansky M, Bozanic N:
SPIKY: A graphical user interface for monitoring spike train synchrony.
Submitted to JNeurophysiol and the arXiv [PDF] (2015)

Bozanic N, Mulansky M, Kreuz T:
SPIKY
Scholarpedia **9**(12), 32344 (2014).

Mulansky M, Bozanic N, Sburlea A, Kreuz T:
A guide to time-resolved and parameter-free measures of spike train synchrony.
Submitted to IEEE and the arXiv [PDF] (2015).

Kreuz T, Chicharro D, Houghton C, Andrzejak RG, Mormann F:
Monitoring spike train synchrony.
J Neurophysiol **109**, 1457 (2013) [PDF].

Kreuz T:
SPIKE-distance.
Scholarpedia **7**(12), 30652 (2012).

Kreuz T:
Measures of spike train synchrony.
Scholarpedia **6**(10), 11934 (2011).

Kreuz T, Chicharro D, Greschner M, Andrzejak RG:
Time-resolved and time-scale adaptive measures of spike train synchrony.
J Neurosci Methods **195**, 92 (2011) [PDF].

Kreuz T, Chicharro D, Andrzejak RG, Haas JS, Abarbanel HDI:
Measuring multiple spike train synchrony.
J Neurosci Methods **183**, 287 (2009) [PDF].

Kreuz T, Haas JS, Morelli A, Abarbanel HDI, Politi A:
Measuring spike train synchrony.
J Neurosci Methods **165**, 151 (2007) [PDF].

For questions and comments please contact us at "thomaskreuz (at) cnr.it".

Input:

Matrix 'spikes' with two or more spike trains (if trains have different numbers of spikes, fill with zeros)

Parameter structure 'para' that describe the data (see below)

tmin: Beginning of recording

tmax: End of recording

dts: Sampling interval, precision of spike times [!!! Please take care that this value is not larger than the actual sampling size, otherwise two spikes can occur at the same time instant and this can lead to problems in the algorithm !!!]

select_measures: Vector with order of measures

Output (Structure 'SPIKY_results'):

SPIKY_results.<Measure>.name: Name of selected measures (helps to identify the order within all other variables)

SPIKY_results.<Measure>.distance: Level of dissimilarity over all spike trains and the whole interval

just one value, obtained by averaging over both spike trains and time

SPIKY_results.<Measure>.matrix: Pairwise distance matrices, obtained by averaging over time

SPIKY_results.<Measure>.time: Time-values of overall dissimilarity profile

SPIKY_results.<Measure>.profile: Overall dissimilarity profile obtained by averaging over spike train pairs

Note: For the ISI-distance the function 'SPIKY_f_pico' can be used to obtain the average value as well as

x- and y-vectors for plotting (see example below):

[overall_dissimilarity,plot_x_values,plot_y_values] =
SPIKY_f_pico(SPIKY_results.<Measure>.time,SPIKY_results. <Measure>.profile,para.tmin);