Documentation for Matlab NEV-processing functions:

LoadNEV Read data from a NEV file

nevVariable = loadNEV(filename, channellist, units, detail) opens a NEV file and reads in all the specified information. Particular channels can be selected and unclassified units can be disregarded. In addition, waveforms and non-neural experiment information can be loaded in.

Note: The data is loaded into memory but will not be physically stored. The data can simply be as a mat-file using 'save filename' to save all variables in the workspace or 'save filename nevVariable' to only save the data that was loaded in.

The *filename* is a string containing the name and possibly the path for the file to be read.

The optional *channellist* contains an array of the channels that are read.

units: The optional string argument 'no' is used to only read classified units. If not specified, all units will be read. Noise will never be read.

The *detail* argument is optional and waveforms or non-neural experiment information will not be read if not specified. Valid entries are:

'all' read waveforms and non-neural experiment information

'wav' read waveforms of recorded neural packets

'exp' read non-neural experiment information

Example: data = loadNEV('sim100.nev', [1:20], 'all'); (This will load in channels 1 through 20 from the file sim100.nev. All waveforms are loaded.)

The data is written into a structure array and is organized in the following manner:

nevObject.

|-.FileInfo – information about the NEV file

|-.source – location and name of the NEV file

|-.packetCount – number of packets in the NEV file

|-.bytesPerWaveformSample – number of bytes per waveform sample

|-.HeaderBasic – basic information about the data in the NEV file

|-.fileTypeID – type of file, usually "NEURALEV"

|-.fileSpec – file specification used to create the file

|-.formatAddtl – file format additional flag (uint16)

|-.dataOffset – number of bytes in the headers

|-.packetLength – number of bytes per data packet

|-.timeResolution – time resolution of time stamps (uint32)

|-.sampleResolution – time resolution of samples (uint32)

|-.timeOrigin – time at which data was collected (Greenwich Mean Time) (uint16)

|-.application – application used to create the file

|-.comment – comment field

|-.headerCount – number of extended headers (uint32)

|-.HeaderExtended – cell array with nevVariable.HeaderBasic.headerCount number of elements; there are two main kinds of extended headers: the settings for neural event channels and the configuration of experiment information channels.

Neural event channels:

|-.electrode – electrode ID number used in the data section of the file (uint16)

|-.module – physical system connector or module connected to the electrode (uint8)

|-.pin – physical system connector pin or channel connected to the electrode (uint8)

|-.scale – digitization scaling factor (nV per LSB step) (uint16)

|-.energy – energy threshold, 0 if none used (uint16)

|-.amplitudeHi – amplitude high threshold used (in uV) (uint16)

|-.amplitudeLo – amplitude low threshold used (in uV) (uint16)

|-.unitCount – number of sorted units in channel, set to 0 for no unit classification (uint8)

|-.bytesPerSample – number of bytes per waveform sample, a value of 0 indicates 1 byte (uint8)

Experiment information channels:

|-.periodicFreq – frequency of periodic packet generation, 0 if none (uint16)

|-.DIOConfig – digital input port configuration (0 if digital input port changes are ignored; 1 if digital input port changes cause experiment information event) (uint8)

|-.Analog1Config – external analog channel 1 configuration (0 - no trigger, 1 - low to high edge triggers experiment information event, 2 - high to low triggers experiment information event, 3 - both low to high and high to low edge triggers experiment information event) (uint8)

|-.Analog1Threshold – analog channel 1 edge detect value in mV (int16)

|-.Analog2Config – external analog channel 2 configuration (0 - no trigger, 1 - low to high edge triggers experiment information event, 2 - high to low triggers experiment information event, 3 - both low to high and high to low edge triggers experiment information event) (uint8)

|-.Analog2Threshold – analog channel 2 edge detect value in mV (int16)

|-.Analog3Config – external analog channel 3 configuration (0 - no trigger, 1 - low to high edge triggers experiment information event, 2 - high to low triggers experiment information event, 3 - both low to high and high to low edge triggers experiment information event) (uint8)

|-.Analog3Threshold – analog channel 3 edge detect value in mV (int16)

|-.Analog4Config – external analog channel 4 configuration (0 - no trigger, 1 - low to high edge triggers experiment information event, 2 - high to low triggers

experiment information event, 3 - both low to high and high to low edge triggers experiment information event) (uint8)

|-.Analog4Threshold – analog channel 4 edge detect value in mV (int16)

|-.Analog5Config – external analog channel 5 configuration (0 - no trigger, 1 - low to high edge triggers experiment information event, 2 - high to low triggers experiment information event, 3 - both low to high and high to low edge triggers experiment information event) (uint8)

|-.Analog5Threshold – analog channel 5 edge detect value in mV (int16)

|-.GeneralInfo

|-.timestamps – array of times at which events occurred (uint32)

|-.packetOrder – array of packet IDs corresponding to timestamps above (uint8)

|-.packetNumbers – sequential numbering of packets, this is used to retain the

|-.unitOrder – array of either unit classification numbers or packet flags (uint8)

|-.NumberSpikes – array of the number of spikes in each channel

|-.ActiveChannels – array of the electrode channels that contain recorded unit activity (only includes channels/units that have been selected)

|-.SpikeData – matrix with all channels/units that have been selected

|-.timestamps – timestamps on the particular channel and unit (uint32)

|-.waveforms – corresponding waveforms on the particular channel and unit (int8 or int16 depending on the NEV file)

|-.ExpData

|-.timestamps – array of all non-neural experiment information (uint32)

|-.flags – array of flag fields detailing why a packet was inserted (uint8)

|-.digital – array of digital input port values (int16)

|-.analog1 – array of analog input channel 1 values in mV (int16)

|-.analog2 – array of analog input channel 2 values in mV (int16)

|-.analog3 – array of analog input channel 3 values in mV (int16)

|-.analog4 – array of analog input channel 4 values in mV (int16)

|-.analog5 – array of analog input channel 5 values in mV (int16)

Note: As indicated above, some data is not stored in ‘double’ precision to save memory and hard drive space. Some Matlab functions require the data to be in double precision in which case it can be converted using the command double(*variable*).

Note: The second index marking the unit number in SpikeData is shifted up by one because Matlab arrays start at 1 instead of 0. This means that unclassified units are indexed with 1, unit one with 2 and so on.

Examples for accessing data:

plot(data.SpikeData(3,2).waveforms) - This plots all waveforms for unit number 1 on channel 3.

data.HeaderExtended{26} - This shows the extended header information on channel 26.

data.GeneralInfo.ActiveChannels - This shows all channels that have units on them (but only on the channels that were selected to be loaded).

plotRaster Plots a raster of the unit activity.

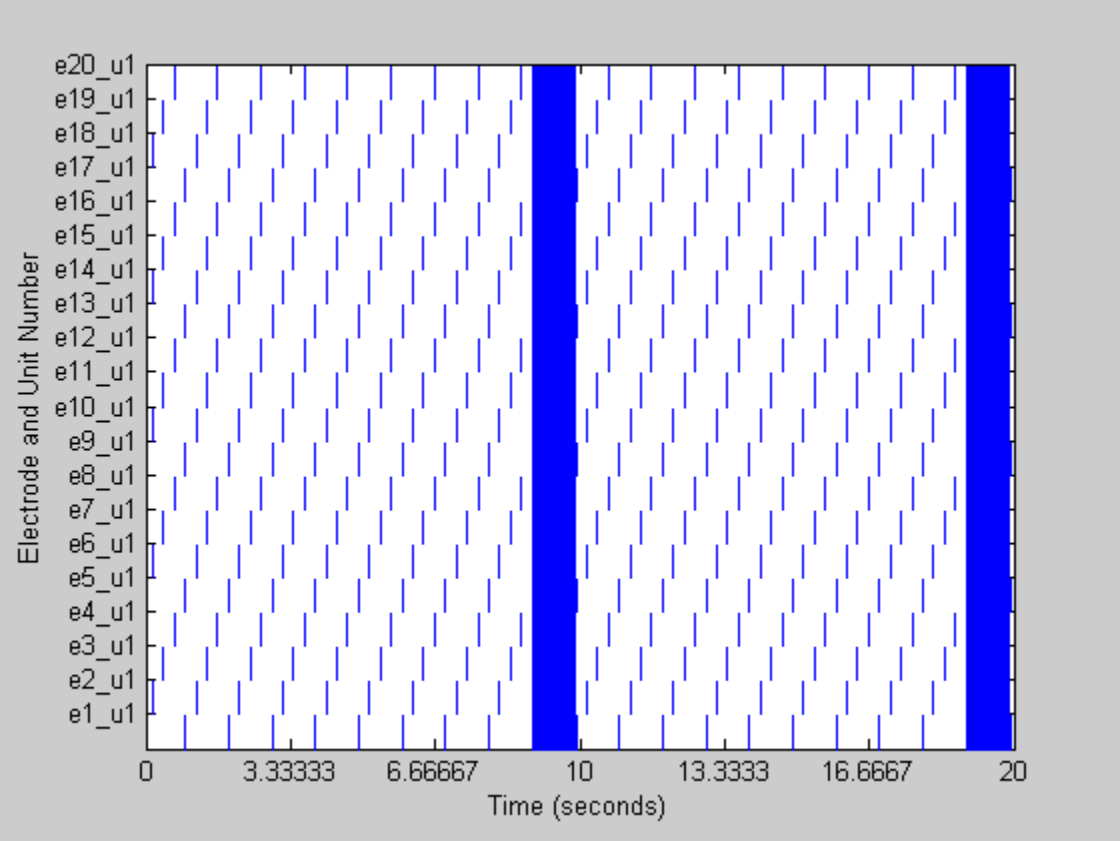
plotRaster(varname, channellist) plots a raster of unit activity. Particular channels can be selected and will be split by defined units.

The *varname* is a string containing the name of the variable into which the nev-file was loaded.

The optional *channellist* contains an array of the channels that are plotted.

Example: plotRaster(data, [1:20]); (This will plot a raster for the activity on channels 1 through 20.)

Note: The figures between channels might overlap on the screen unless the window is expanded. It will print non-overlapping on letter size paper.



Note: This example was created with the sim100.nev file (unclassified units).

plotUnits Plots the units on each channels.

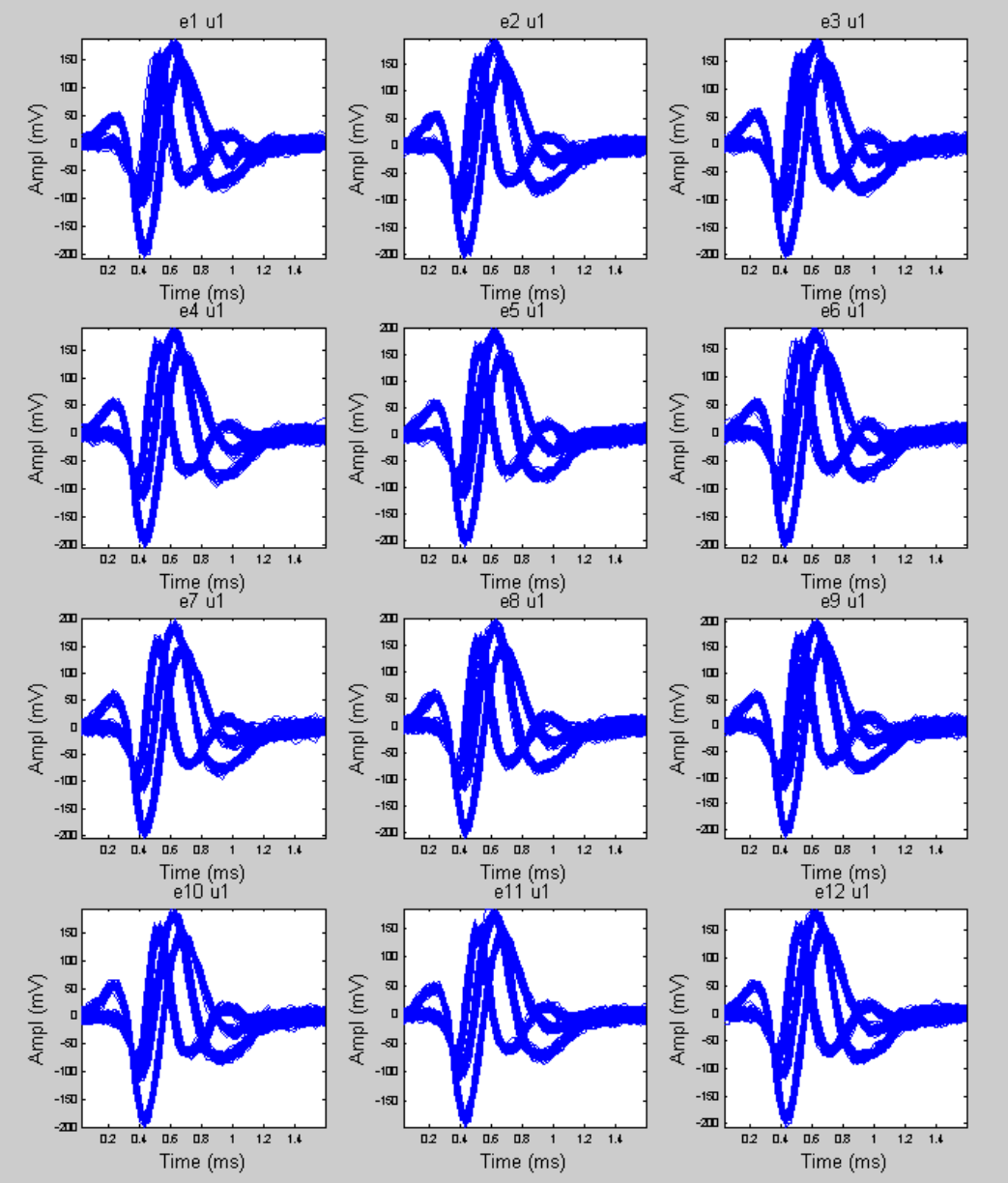
plotUnits(varname, channellist) plots the waveforms for the units on the selected channels. Particular channels can be selected and will be split by defined units.

The *varname* is a string containing the name of the variable into which the nev-file was loaded.

The optional *channellist* contains an array of the channels that are plotted.

Example: plotUnits(data, [1:20]); (This will plot the waveforms of the units on channels 1 through 20.)

Note: The figures between channels might overlap on the screen unless the window is expanded. It will print non-overlapping on letter size paper.



Note: This example was created with the sim100.nev file (unclassified units).

LoadCont Read data from a NEV file and recreate a continuous signal

nevVariable = loadCont(filename, channellist, detail) opens a NEV file and reads in all the specified information. Particular channels can be selected and non-neural experiment information can be loaded in. The data packets (waveforms) on a channel are concatinated to recreate a continuous signal.

Note: This function can only be used with unclassified NEV-files!

If thresholds are set to 0 (Low) and 1 (High) on the 100-Channel Data Acquisition system, continuous data can be recorded on up to 25 electrodes depending on the computer used.

Do not the loaded data with the *plotRaster* or *plotUnits*!

The data is loaded into memory but will not be physically stored. The data can simply be as a mat-file using 'save filename' to save all variables in the workspace or 'save filename nevVariable' to only save the data that was loaded in.

The *filename* is a string containing the name and possibly the path for the file to be read.

The optional *channellist* contains an array of the channels that are read.

The *detail* argument is optional and non-neural experiment information will not be read if not specified.

'exp' read non-neural experiment information

Example: data = loadCont('sim100.nev', [1:20]); (This will load in channels 1 through 20 from the file sim100.nev.)

The data is written into a structure array and is organized in the same manner as with loadNEV with the following exception:

nevObject.

|-.SpikeData – vector for all channels that have been selected

|-.timestamps – timestamps on the particular channel (uint32)

|-.waveforms – continuous signal on the particular channel (int8 or int16 depending on the NEV file)

Note: As indicated above, some data is not stored in 'double' precision to save memory and hard drive space. Some Matlab functions require the data to be in double precision in which case it can be converted using the command double(*variable*).

plotCont Plots the continuous signals on each channel.

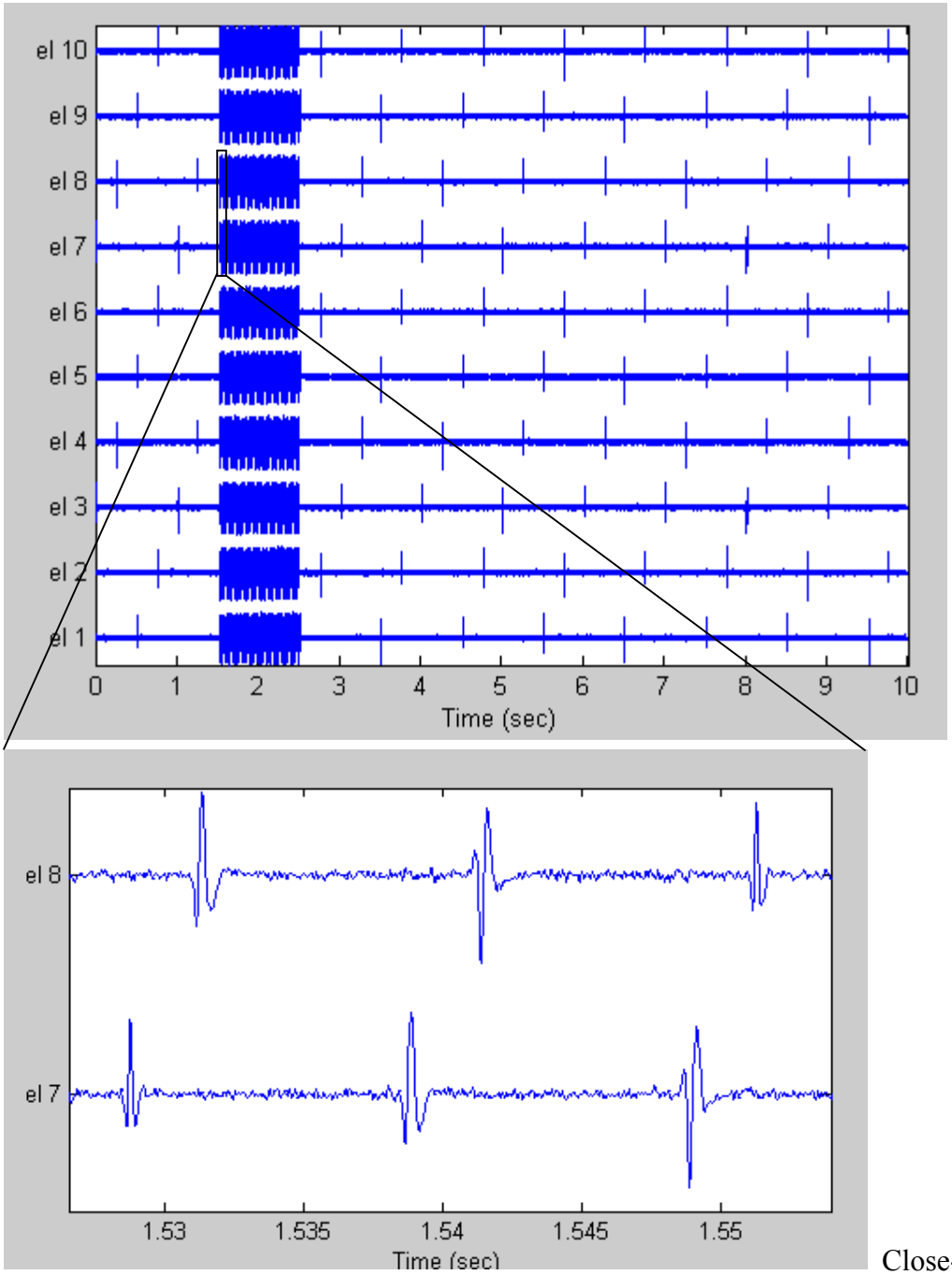
plotCont(varname, channellist) plots the continuous activity for the selected channels.

The *varname* is a string containing the name of the variable into which the nev-file was loaded.

The optional *channellist* contains an array of the channels that are plotted.

Example: plotCont(data, [1:20]); (This will plot the activity on channels 1 through 20.)

Note: If data packets are missing in between, Matlab will draw a line between the last and the next packet.



Close-up of some units