

“p_pod”
a MATLAB program for pre-processing of EROS data
Ed Maclin, Jeff O’Dell, Kathy Low
And a large team of “beta” testers

The program “p_pod” is a graphical user interface (GUI) to a batch processor that calls selected subroutines to perform various processing steps on data collected by ISS Corporations “boxy” program, ultimately resulting in stimulus or response locked averaged data files suitable for use with Gabriele Gratton’s program “opt3d”.

p_pod can perform five main processing steps; normalization, pulse correction, filtering, averaging and conversion of the averages to opt3d format. These steps are accomplished by the following subroutines:

- b_norm.m** - reads ISS “Boxy” data files, 'normalizes them', removes trends in phase (drift) and corrects phase wrap and outliers. It saves normalized data in arrays named "dc", "ac", "ph", extra 'boxy' variables 'mark', 'boxy_flags', 'aux1', 'aux2' and 'digaux' and the means of the raw data as "mrdc", "mrac", and "mrph" in .mat format files in the directory “norm00-00”.
- e_pulse.m** - Removes pulse artifact from unfiltered normalized data per Gabriele's algorithm. e_pulse saves various heart rate related data in a “HR” file and optionally plots intermediate results.
- e_filt.m** - Bandpass filters either "normalized" or "pulse corrected" data using a two-pass (zero phase shift) Butterworth filter.
- e_avg.m** - Epochs and computes event locked averages in bins defined by .evt files, which are generated by a separate program (see below).
- eo3d.m** - Writes averaged results in the format required by “opt3d”.

Each of these processing stages can be selected in the p_pod GUI by setting the appropriate button (see below).

P_pod_v10_beta Release notes (April 19th, 2013)

This release comprises numerous changes which make p_pod incompatible with data files produced by previous versions. Older datasets will have to be reprocessed starting with the original (renamed) “boxy” files.

The main changes are:

- 1) Data acquisition parameters are read directly from the boxy file and saved in output files as ‘boxy_hdr’.
 - 2) Auxiliary data (time group step mark flag a2d and digaux are read from boxy file and saved in output files.
 - 3) Quality Control procedures are applied at both normalization and averaging stages, if desired.
 - 4) A ‘hdr’ structure with all current parameters is saved in output files at every stage of analysis.
 - 5) The user now must select both an ‘event’ folder and an output folder when averaging.
 - 6) The GUI has been modified to reflect the above changes and additional small issues.
- 1) The following acquisition parameters are read from each original boxy data file at normalization and saved in all output files as a structure named ‘boxy_hdr’: n_dets, n_srcs, n_chans, n_a2d_aux, CCF, n_skipped, n_waves_avg, n_cycles_avg, n_acq_per_wave, n_dig_aux, sample_rate, period, period_ms, n_points).
 - 2) The following auxiliary data are read from each boxy data file and saved in all output files as a structure named ‘aux’: time, group, step, mark, flag, aux_a2d(1-n), digaux(1-n).
 - 3) Several “quality control” procedures are applied during pre-processing. These are checks that various aspects of the data are within acceptable ranges. During normalization, points in the phase data that are more than 3 Standard Deviations from the mean are set to the mean of the previous and subsequent points.

During averaging, six criteria are applied:

- a. the “Mean Raw DC” (mrdc) and
- b. the “Mean Raw AC” (mrac) on each channel, computed across the entire block, must be above appropriate thresholds. If not, the channel is marked as “bad” for all epochs
- c. The DC “range” (max-min) within an epoch must be less than a specified threshold
- d. The phase (delay) range within an epoch must be less than a specified threshold
- e. The mean modulation (mrac/mrdc) across each epoch must be above a specified threshold

The number of epochs accepted by each test, and the total number accepted are recorded in a .xlsx file for each channel, permitting post-hoc detection of problematic channels.

- 4) The ‘hdr’ structure is saved as ‘norm_hdr’, ‘pc_hdr’, ‘filt_hdr’ and ‘avg_hdr’ at each corresponding stage of analysis, along with headers from every previous stage making it possible to reconstruct all the pre-processing parameters applied to the final .avm files.
- 5) When averaging, the user is prompted for both the folder containing the event files that will be used and for the folder where the .avm and .avg files will be stored. This allows both for different event types (e.g. stimulus vs. response locked) and different quality control settings to be stored separately.
- 6) The following changes have been made to the GUI:
 - a. The acquisition parameters menu has been removed, as these are read from the boxy file. The “number of blocks” parameter is now entered on the “batch” menu.
 - b. Initial quality control parameters are entered on the “Parameters” menu, and Average Quality Control parameters are entered on “Parameters:Analysis”.
 - c. Check boxes to choose whether or not to apply initial or avg quality control parameters have been added to the main window.
 - d. The “Select Channels” checkbox has been removed. All channels are now processed and written to the .avg files.
 - e. An “Edit .skp files” button has been added.

Required Files:

Several files are also required by p_pod, and these must be in specific directories as described below. See appendix B for file formats:

- 1) a "header" file (e.g. "*.hdr") which defines various parameters,
- 2) a "skip" file (.skp) which flags blocks to be skipped. This file contains an array of 1s and 0s, where 1 indicates a good block and 0 a bad block; each row represents a configuration, each column a block.
- 3) a modulation frequency (.mdf) file that defines the modulation frequency used in each block
- 4) an .elp (Polhemus output) & .mtg file per subject; a .dis OR .gch as well. These files are required for channel selection (see below).
- 5) an "event" file which contains a list of event labels and the time at which each event occurred.

Event file generation:

An additional program is required to generate the "event file". This program is unique to each experiment, though the general form should be similar across experiments run using a particular stimulus generation program. An example program is "**arc_sto_Stimlocked.m**" which reads pc-exp's .sto files from the ARC experiment, or "beh2evt_psd.m" which reads an eprime excel output file and generates the event file.

Data Directory Structure:

The data are organized in the following directory structure, which **MUST** be used in order for p_pod to work. "... can be a local disk (e.g. "c:") or a network drive (e.g. "\\cni-darwin"):

```

... \data\ "experiment name" \
  opt      contains original (boxy) data files, .skp and .mdf files
  beh      contains the original "behavioral" files from pc-exp, eprime or wherever.
  evt      contains "event" files for each block
  reg      contains .mtg, .elp and (optionally) .dis, .gch and *_good.mtg files
  norm00-00 contains normalized (unfiltered) data and corresponding averaged data
  normnn-mm contains filtered but NOT pulse corrected data, and corresponding averaged data
  PC00-00  contains unfiltered pulse corrected data and corresponding averaged data
  PCnn-nn  contains filtered, pulse corrected data and corresponding averaged data

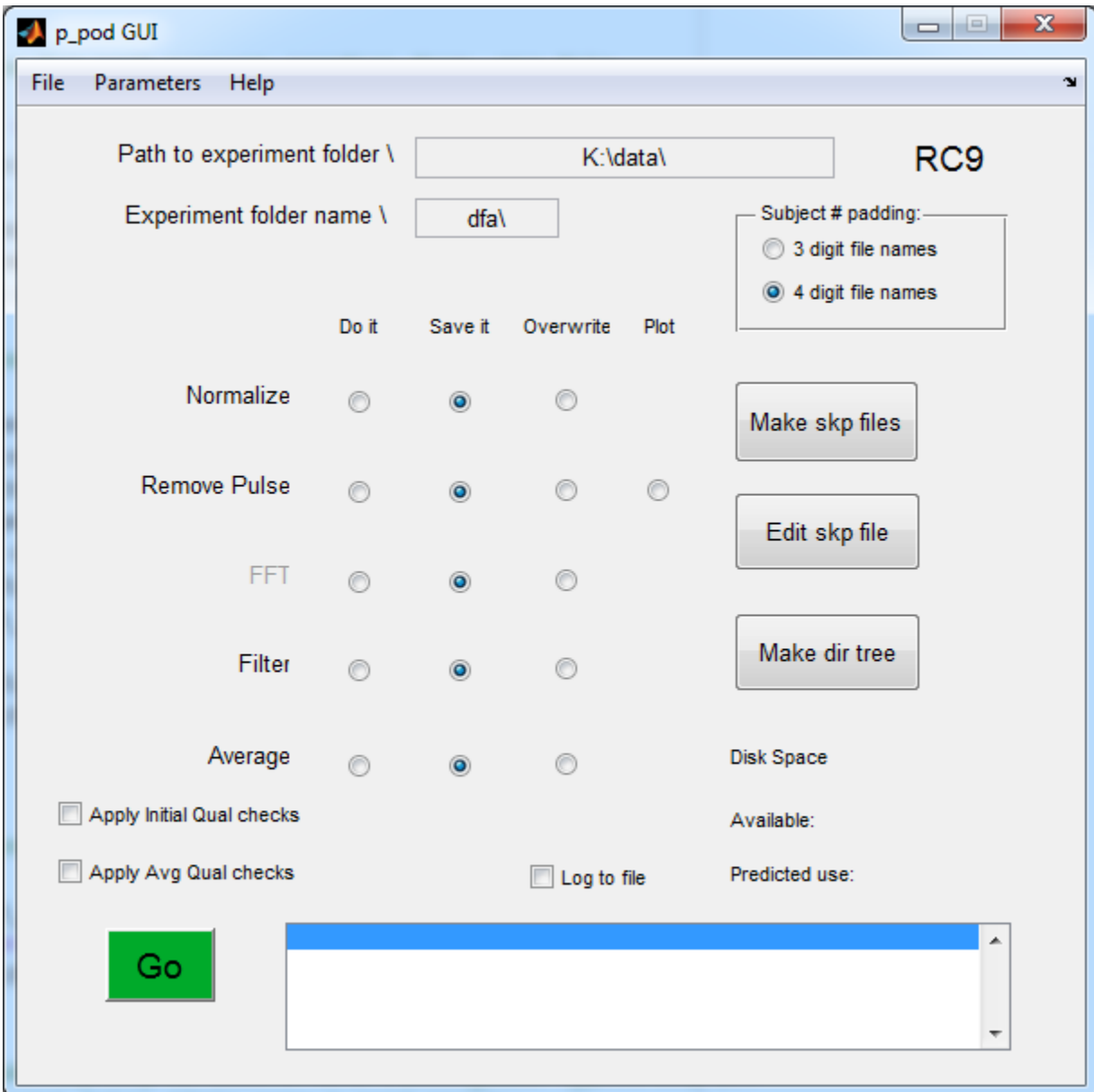
```

File Name Convention (all file names are must be structured as follows):

SS+subj+cf where SS is the "session label", subj is the (zero padded 3 digit) subject number, and cf is the montage label. These variables are defined in the batch parameter menu. The "session label" is usually simply the 3 letter experiment name, but can also be used to discriminate subsets of experimental sessions.

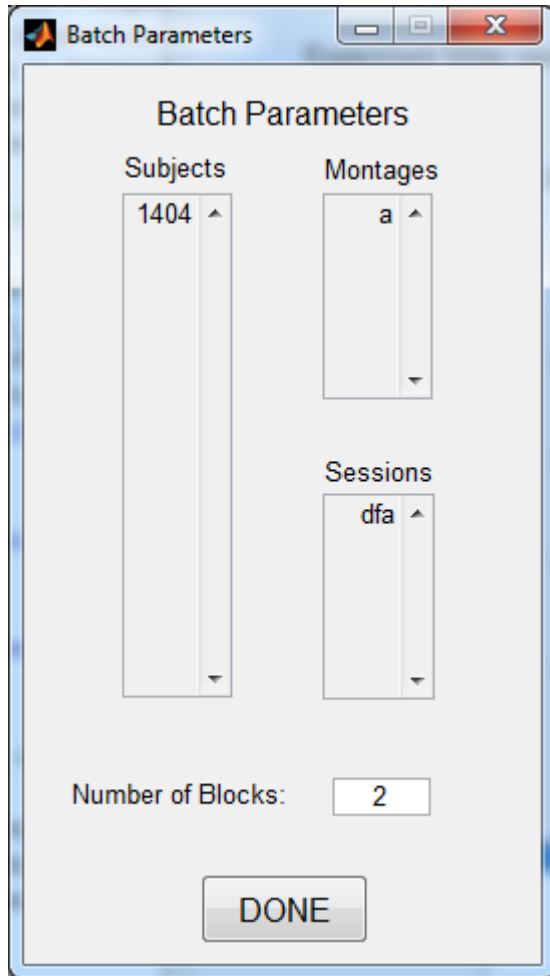
Running p_pod:

To use p_pod, you need to set ALL the parameters and paths to the appropriate values for your experiment, and then SAVE these values as a “header” (.hdr) file. Then you will select the functions you want to perform by clicking the appropriate “function buttons”. To actually do anything, you press the “GO” button.

**Setting Parameters:**

1) Enter the “root data path” in the “Path to experiment” field. This will normally start with the device name (e.g. “c:” or “d:”) and must include everything in the path up to the name of the experiment folder (e.g. “\data”). Enter the experiment abbreviation in the “Experiment Folder Name” field.

2) Click the "Parameters menu", then "Batch". You should see:

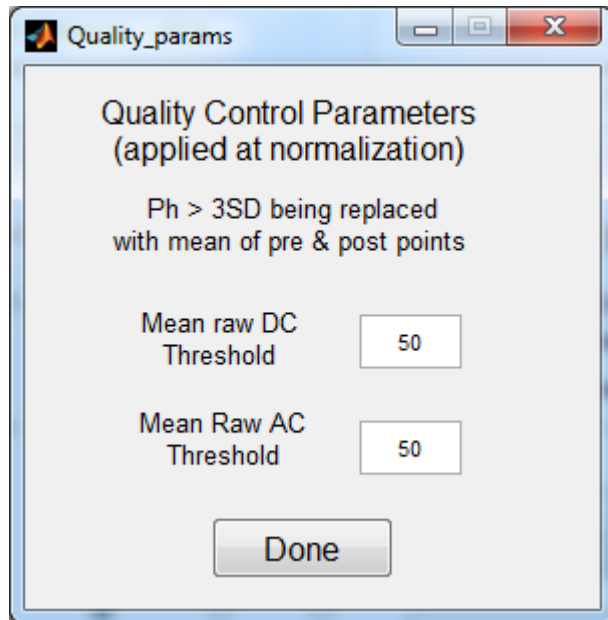


These parameters control what data files p_pod will analyze. P_pod will loop through all the subjects, montages, sessions and blocks listed here.

File names are structured as follows:

SS+subj+cf.nnn where SS is the "session label", subj is the (zero padded 3 or 4 digit) subject number, cf is the montage label and nnn is the 3 digit padded block number.

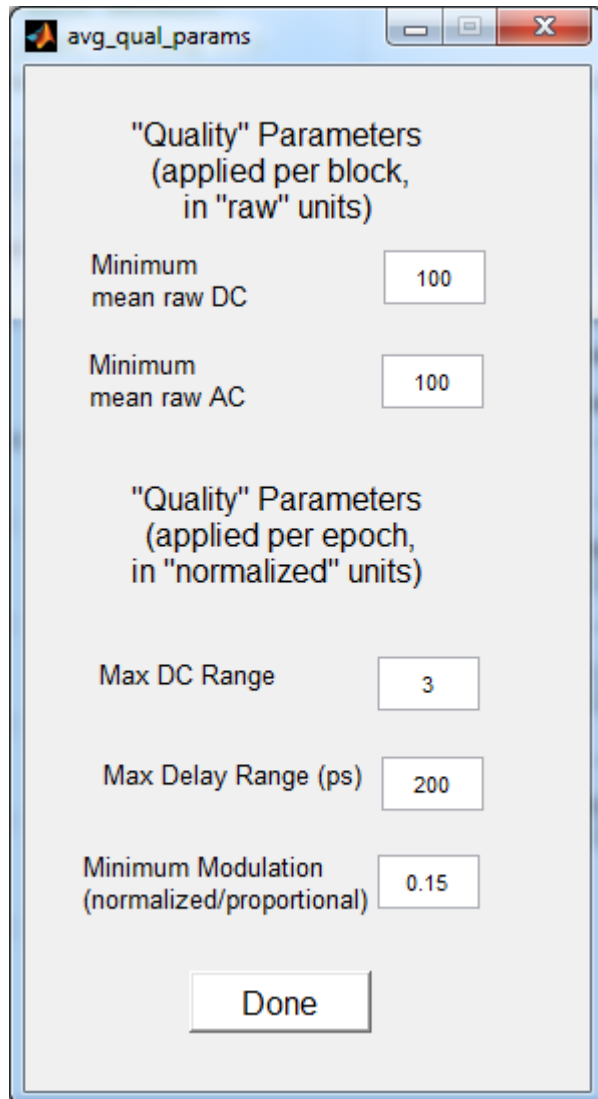
4) Click the “Parameter menu”, then “Analysis”, then “Initial Quality params”. You should see:



The maximum deviation from the mean phase (which has been “normalized” to zero) which will be accepted defaults to 3. Points exceeding this value are set to zero.

The “mrdc” and “mrac” (Mean Raw DC & AC) thresholds set here determine what channels will not be processed in any subsequent processing (e.g. pulse correction, filtering). This is primarily used to speed up processing by eliminating extremely noisy (low signal) channels. More stringent criteria for these parameters can be applied to each epoch during averaging.

4) Click the "Parameter menu", then "Analysis", then "Avg Quality params". You should see:



The screenshot shows a dialog box titled "avg_qual_params" with the following parameters:

Parameter	Value
"Quality" Parameters (applied per block, in "raw" units)	
Minimum mean raw DC	100
Minimum mean raw AC	100
"Quality" Parameters (applied per epoch, in "normalized" units)	
Max DC Range	3
Max Delay Range (ps)	200
Minimum Modulation (normalized/proportional)	0.15

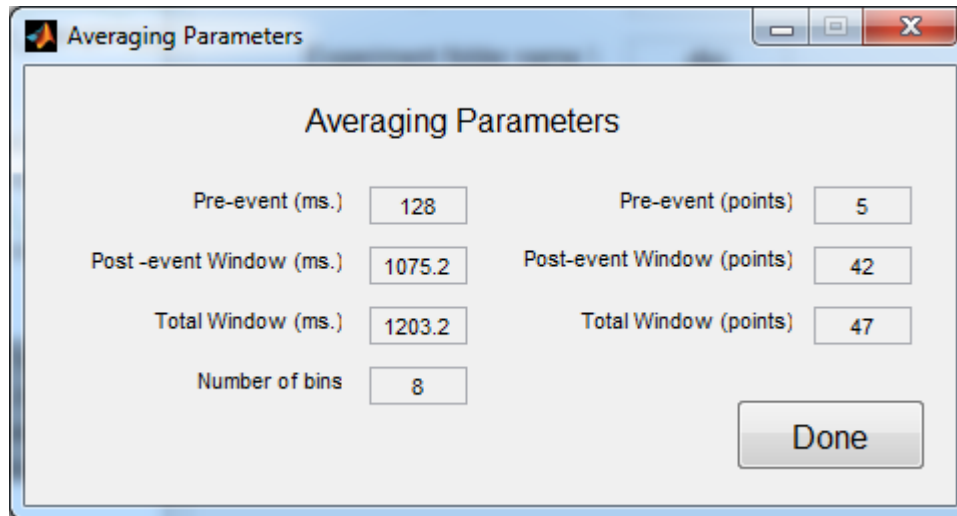
A "Done" button is located at the bottom of the dialog.

The "Minimum raw DC" and "Minimum mean raw AC" thresholds are specified in original units ("counts"). Channels whose mean across the entire block is LESS than these values will be ignored for ALL epochs.

The "Max DC Range" sets the maximum acceptable range of DC within each epoch.

The "Max Delay Range" sets the maximum acceptable range of delay within each epoch.

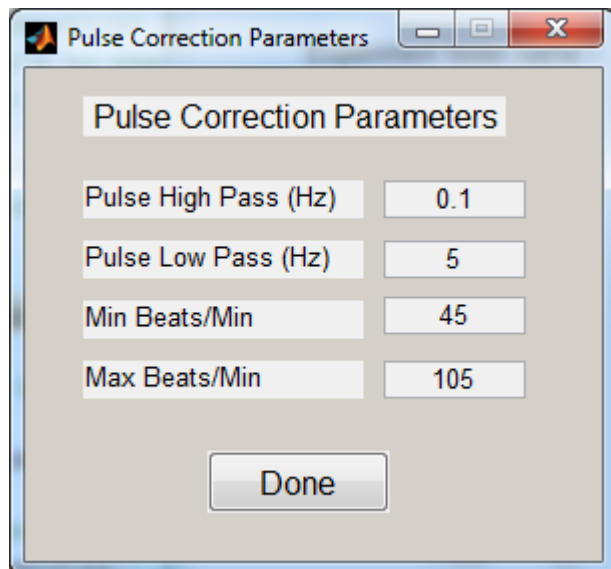
The "Minimum Modulation" is applied the mean (across the epoch) of RawAC/rawDC. This value is specified as a proportion, not a percent.



These parameters control how the data will be epoched (segmented) and averaged. The pre-event, post-event and total window values can be entered either in milliseconds or in data sample points. The corresponding values in the other units will update automatically. Changing the pre- or post- event values will cause the total value to change accordingly. Changing the total value will update the post event window. The point referenced in the event file becomes the LAST of the “pre-event” points. This means you should enter the number of “pre-event” points in Opt-3D’s “point used for zero value”. These parameters do NOT need to be set up prior to normalizing or pulse correction, but need to be set prior to averaging. The “Number of bins” can be ignored, as this is actually inferred from the event file.

4) Click the “Parameters menu”, then “Analysis”, then “Pulse Correction”. You should see:

These parameters control various aspects of the pulse correction process

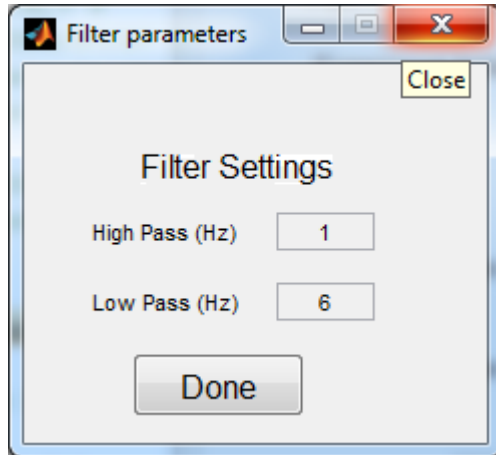


The high and Low pass settings determine how the data are filtered prior to beat detection. These normally will not need to be changed.

The Max and Min Beat/Min restrict the range of heart rates that will be scanned. These may need to be adjusted for subjects with unusual heart rates (or variability).

The Warp Size is now automatically set to 2 * the number of points in the maximum beat interval (minimum beat frequency).

5) Click the “Parameters menu”, then “Analysis”, then “Filter”. You should see:



You can set the high and low pass frequencies for a 6th order butterworth band pass filter here.

Entering a value of “0” for the High or Low Pass will produce a “low pass” or “high pass” filter respectively, while if both values are >0 a band pass filter is produced.

Now that all the parameters are set, click on the “File Menu”, and “Save As”. Save the header file in your experiment folder. The next time you run you can simply use the “Open file” choice to load this parameter file and it will set all the parameters for you.

Selecting Processing Steps:

To actually process the data, you need to first select which operations you wish to perform, and what type of data (pulse corrected or not, filtered this way or that) by setting the appropriate “function” buttons. It is important to recognize that if the files produced by any requested analysis step already exist, it is still necessary to set the button in order to indicate what type of data should be used for subsequent steps. Say, for example, that you have a data set that you have pulse corrected, filtered and averaged with event files locked to the stimulus, and now you want to produce response locked averages. Set the “Remove Pulse” and “Filter” (make sure the filter parameters are correct) and “Average” buttons. This tells p_pod that you want to average the pulse corrected and filtered data that already exist.

The interaction of the “Normalize” and “Remove Pulse” buttons may not be intuitive. If either the “Normalize” OR “Remove Pulse” button is selected, just that type of data will be processed. If neither is selected, but the filter and/or average buttons are selected, the normalized data will be processed by default. If BOTH are selected, the pulse corrected data will be processed.

When averaging, the user is prompted to select both an “event” folder, which must exist and contain the appropriate event files, and to select (or create) a folder to put the .avm, .avg, .nirs and .xlsx files in. The prompt defaults to the same directory the input data comes from (e.g. norm00-00 or pc01-15); in order to keep track of how the data was processed the user should choose or create a folder that identifies the type of events and quality parameters being used.

The “save it” and “Overwrite outputs” buttons for each function allow you to control the creation and preservation of each functions output files. The default is to attempt to save all files, but not to overwrite existing files. You might decide that you don’t need to keep a copy of the normalized data, because you will always want to use pulse corrected data. In this case you could unselect the “save it” button on the “normalize” function, and set the “Do it” buttons for both “Normalize” and “Remove Pulse”. Then, when you hit “Go”, the data will be normalized and pulse corrected, but only the pulse corrected data will be saved. The “Overwrite outputs” buttons are useful, for example, if you recognize that some parameter was incorrect in a previous analysis and you want to rerun the analysis but don’t want to manually erase the “bad” files.

Logging:

Checking ‘log to file’ creates a log file in the experiment folder that is of the format: ‘EXP-yyyy-mm-dd.log’ and is automatically appended to for successive runs on the same day.

Disk Space Available:

P-Pod now displays the available disk space left at startup and whenever it writes a file. In the future the predicted use field will be implemented.

Appendix A - File Formats:

P-Pod uses two types of files, ASCII and binary. ASCII files can be read and written using any editor (Notepad, word, etc.); they must be saved as “.txt” files. The ASCII files include the original boxy files, the skip files, the behavioral files (usually), the .elp .src, .det, .dis and .gch files, and the .avg files which are used as input to opt3d. All other files, including the normalized, pulse corrected, filtered and .avm files are binary. Specifically, these files are in Matlab “save” format, and can only be read by using the “load” command in Matlab.

“skip” (.skp) files:

A .skp file (3 sessions of 30 blocks; the last line is an optional aid to counting the blocks):

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
```

Modulation Frequency (.mdf) Files

The modulation frequency file contains the frequency of the light emitted by the oximeter for your experiment per configuration, in simple scientific notation per line. Fox example, if your experiment has 4 configurations where the first and third was run at 300 MHz while the second and fourth at 220 MHz, the .mdf file would look like this:

```

3.0e8
2.2e8
3.0e8
2.2e8
```

Appendix B – Boxy parameter windows

Panel that Controls the Recording Window

Record = data point or sampling point

This will usually be set to 1 because we will re-trigger Boxy with every block in PC-EXP.

This will be determined by the timing of a single block in PC-EXP or eprime. "File fill time" should be longer than the time it takes to complete 1 experimental block.

To calculate, you need to know:

1. the block duration (including the initial baseline period for ERP recording)
2. the sampling period, based on final sampling rate of the optical recording

Ex. Optical sampling rate = 62.5 Hz ; Block duration = 3.3 min
 Sampling period = $1000/62.5 = 16$ ms
 $12600pts * 16 = 201,600$ ms = 201.6 sec = 3.36 min
 →optical will collect data continuously for 3.36 minutes

Record Settings

Number of Record Groups Per Set (1- 5120)

Number of Records Per Record Group (1-100000)

Total records in set 3925
 File fill time (min.) 1.67

Action after each group:

Auto Restart Acq. After Each Group

Pause Acq. After Each Group

Pause Recording After Each Group

Action on completion of set:

Overwrite Data When Complete

Save Data When Complete To auto save the data file

Auto Restart Recording When Complete

Apply

OK Cancel Help

Panel that Controls Data Acquisition Timing Parameters

Timing/Averaging Settings

Waveform (Cross Corr.) Freq. (Hz)

Number of ADC Chns. (Detectors)

Number of Waveforms to Skip

Number of Ext. MUX Chns. (Sources)

Number of Waveforms to Average

Number of (Data Collection) Cycles to Average per Data Update Cycle

Max. Num. of Data Cycles Per X-fer BIK

Number of Aqs. per Waveform (in FFT)

Pack Mult. Data Col. Cycles (in Empty Ext. Mux. Chn. Slots)

AC and Phase can only be measured with 4 or more aqs. per waveform

Master Clock Frequency (Hz) (Standard for sample rate clock) If checked, program will wait for trigger

Use Ext. Trigger

Light Modulation Frequency (Hz) Ext. Trig. For Restart

"Waveform" refers to 1 cycle of the CCF

Number of samples/cycle of the CCF

Will change ADC Sample Rate, Divisor, and DMA rates

Determines how often the data are transferred to memory (X-fer = Transfer)
 Changes DMA Buffer and DMA cycle rate
 Adjust so that DMA cycle rate is b/w 5-10 Hz

	REQUIRED	MAXIMUM	MINIMUM
DMA Buffer (bytes)	61440	524288	
ADC Sample Rate (Hz)	50000	50000.0	
Divisor for sample rate clk.	40.000000000	<i>(Per 4 ADC Channels Scanned)</i>	
		<i>(Must be an integer)</i>	
DMA Cycle Rate (Hz)	6.5105	300.0000	0.0500

Data Collect. Cycle Rate / Period 39.0625Hz / 0.0256s

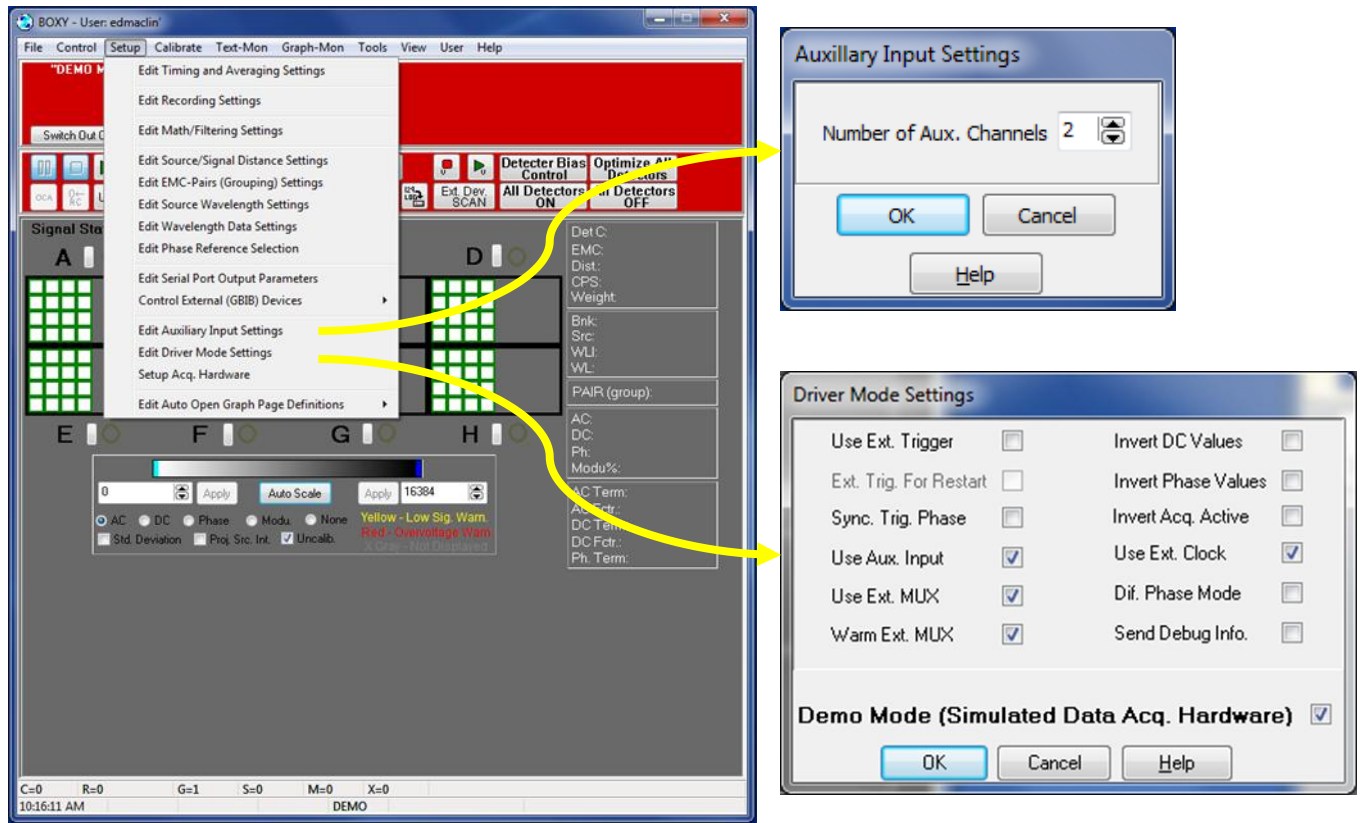
Data Update Rate / Period 39.0625Hz / 0.0256s

Data Update Rate = final sampling rate

The final sampling rate is determined by:

1. the number of sources (mux channels)
2. the number of waveforms to skip
3. the number of waveforms to average

Panels that Control Auxiliary Input Channels



Set the NUMBER of auxiliary A/D (analog) channels in the “Auxiliary Input Settings” window

Enable (both analog and digital?) auxiliary inputs in the “Driver Mode Settings) window.