CITest Documentation Version 1.21, Jan 15, 2015

Overview

CITest is a MATLAB graphical user interface (GUI) designed to control cochlear implant stimulation during a psychophysical or electrophysiological experiment. It currently works with Advanced Bionics implants via the AB research interface hardware and BEDCS software. CITest is modular in design, making it fairly straight-forward to add new experiments or modify features of existing ones. The program is a major rebuild and extension of an earlier smaller application, called "CI_Test", originally written by Julie Arenberg at the University of Washington and later modified by Alex Billings and others at the MRC Cognition and Brain Sciences Unit in Cambridge.

Steven Bierer [sbierer@uw.edu]

Table of Contents

New In This Version :: A renewed sense of possibility

++ The folder "CITest_BEDCS" has been renamed to "BEDCS files" and is intended to be kept in the main directory of the current version of CITest. This change makes it easier to update or swap out the ".bexp" definition files, without affecting the reliability of older CITest versions. See the Installation section for more details.

++ Experiments are now chosen from a pull-down menu, *MAIN TYPE*, rather than from a set of radio buttons. This is to facilitate expansion to more experiment types (ah, the possibilities!).

++ Psychophysical tuning curves is now one of the choices for *MAIN TYPE*. This experiment sets up a 2-channel masker-probe stimulation paradigm, intended for the construction of tuning curves. See Experiment Types for details.

++ Additional options for setting up experiments are now available under the *SUB TYPE* pull-down menu in the EXPERIMENT TYPE panel. This will allow, for example, an experiment to be run using either standard biphasic pulses or pseudomonophasic pulses. Another option is biphasic single pulses, which is now a standard subtype choice for threshold and MCL experiments.

++ Pulse shape parameters can be viewed and edited using the new Pulse Modification menu, which can be opened by pressing the button labelled *PULSE MOD* in the STIMULUS PARAMETERS panel. This menu replaces the "1-Pulse" check box from version 1.13.

++ "3-Interval Forced Choice" has been added to the list of runtime modes for all experiment types. It works basically the same as the previous 2-IFC mode, and uses mostly the same code.

++ Button functionality for the IFC runmodes has been improved. Previously, a thin border would appear around a button after it was pressed, a change that persisted into the next stimulus trial. The code now properly resets the buttons' Java property that was causing this problem.

++ The Channel Selector popup window now opens much faster (the renewal part!). The code was improved to make it a more generic tool, so that more than one channel can use the same GUI interface while maintaining seperate channel parameters (e.g. for the masker and probe channels defined by the new psychophysical tuning curve module).

++ This version lays the groundwork for a more advanced Results View window that can hold information about previous runs as well as "push" certain results (like threshold or loudness balancing results) to the Channel Selector menu. Most of this functionality, however, won't be implemented until a later version of 1.2.

++ Data analysis routines can now be executed after a CITest run by pressing the *ANALYSIS* button on the Results View Window. See Results View Window for details.

++ “dB” is now an accepted keyword when filling out UI fields for current level. Expressed with a number, it simply converts the input string from dB re: 1µA to µA. For example, the string “20 dB” is automatically and immediately rewritten as “10.0” (µA). It can also be used in combination with other keywords, such as “*thr + 2dB*”.

++ A separate BEDCS definition for the the monopolar (MP) electrode configuration was created as a special case for pTP and sQP. This increases the total number of allowable active electrodes when sigma = 0.

++ The bipolar (BP) electrode configuration is now fully operational for all experiment types.

++ Starting and maximum levels, which use to be reset to an arbitrarily low level when certain stimulus parameters were altered, will no longer be changed if the setting is based on a relative level like threshold or MCL. Safety checks will still be performed at the time the *START* button is pressed.

== Information about features introduced in earlier CITest versions may be found in the older documentation files (e.g. CITest_Documentation_v01.13.docx). ==============================

Installation

Follow these instructions when setting up CITest for the first time. Some steps may also be required when upgrading to a new version.

- Make sure the BEDCS application (version 1.18), and MATLAB (R2007B or later, though older versions may also work) are properly installed on your computer.
- Copy or unzip the folder containing the newest version of CITest onto your own computer and add it to the directory path within the MATLAB application. This folder contains all the m-files necessary to run CITest.
- The subfolder “BEDCS files” contains BEDCS experiment definitions that are called from CITest. It can be left in the main CITest directory (recommended) or moved to another location (such as a generic folder for other BEDCS experiment files used by that computer) and optionally renamed. If the latter option is taken, the user should edit the variable `handles.bedcsinfo.dir` in **CITest_UserSettings.m** to the full directory name.

- Open **CITest_UserSettings.m** in the MATLAB editor and look for your lab group's entry within the 'switch/case' blocks of code at the top of the m-file. The string following the 'case' keyword is hereby referred to as the "user ID". (If you don't have an entry, make one by copying another block and creating a new user id.) This m-file contains user-specific instructions for CITest, such as where to look for supporting files and where to position GUI windows on the monitor. Before running CITest for the first time, change the following settings within your block:

  - `handles.fileinfo.directories.top` : This string should be the top-level directory where results from CITest experiments should be kept. Separate folders for individual subjects (and optionally different experiment types) are maintained inside this directory. Specific the entire path name.

  - `handles.fileinfo.directories.threshold` and similar : These strings will instruct CITest to place results files first inside an experiment-specific subdirectory within the top-level directory, THEN inside the subject folder. The string should be the name of the desired subdirectory (not a full path name) for the experiment type designated by the final structure field of the string variable (e.g. '.threshold').

  - `handles.bedcsinfo.dir` : This string designates the partial (default) or full path name of the directory that contains the CITest BEDCS experiments (.bExp files).

  - `DEMOMODE` : Set this to 'true' if you plan to run CITest in demonstration mode, which is useful to run and debug CITest in the absence of a research interface or implant. This mode allows the BEDCS application to be visible, which otherwise must be closed during implant stimulation. For normal operation, set `DEMOMODE` to 'false'.

  - `IDSTR` : Set this string to whatever you want yourself or lab to be named. For example, I call mine 'Steve''s laptop computer' (the two consecutive single quotes are interpreted as a single quote by MATLAB).

- Open **CITest.m** in the MATLAB editor. Scroll down a bit to the beginning of the `CITest_OpeningFcn()` function and find the variable named `COMPUTERID` (~line 54). Set this variable to the user ID string in CITest_UserSettings.m that corresponds to your computer's setup.

- After you've run CITest, you may wish to re-position the main GUI or subject interaction GUI windows. This can be done in **CITest_UserSettings.m** by changing the `handles.xydefault` entries associated with your user ID. These vectors specify the x and y positions of the respective window's bottom-left corner, in character units. The different entries are:

  - `xydefault.main` : The main CITest "control" window. In dual monitor setups, this is typically positioned to appear on the first monitor.

  - `xydefault.runsubject`: The experimental “runtime” window for cases in which the test subject is supposed to interact with the GUI. For this case, the runtime GUI will often be centered on the second of two monitors if available.
  - `xydefault.runcontroller`: The runtime window for cases in which only the controller requires access. A typical placement might be just next to the main GUI.
  - As the number of available experiment types increases, a greater level of customization over window positioning might be warranted. For now, this method seems to work.

- There is one more user setting, `BEDCSDELAY`, which is intended to specify the minimum combined processing time of BEDCS and MATLAB when the implant delivers a series of pulse trains. Knowing this value is useful for CITest to constrain the expected rate of pulse train delivery, so it doesn’t attempt to go faster than the hardware will allow, (sorry, Scottie, you really can’t push it any faster!). Presently, this only matters during Bekesy-style tracking and only for the rate of pulse TRAINS and not of the pulses themselves, whose DSP-programmed timing should be very accurate. The present setting of 80 ms is only an approximation and has thus far been tested only with simple repeated pulse trains of the kind that can be generated with the threshold experiment. Raise this value if inter-stimulus intervals appear to be longer than CITest is letting on. [If more careful calibration is required, let me know and I can help you figure out the best way to do this on your own computer.]
  - Note that the demo mode of operation assumes that the portion of processing delay due to BEDCS is occurring even though the hardware is not connected. So, stimuli will be delivered a bit faster compared to the CITest settings. Also keep in mind that because MATLAB will generally not run in true real time, it is subject to operating system and other types of interrupts that can slow execution down at sporadic intervals. This may cause some randomness in the timing of consecutive pulse trains for experiment modes like Bekesy tracking, for which the stimulus parameters are updated between trains. Pulses within a train should be fine, as that timing is controlled by the specialized AB hardware.

If there is any uncertainty about the validity or syntax of a setting in CITest_UserSettings.m, refer to the user entries for ‘BiererLab’ (standard mode) or ‘BiererLaptop’ (demo mode) for examples. Also, when upgrading to a new CITest version, port over everything from your copy of CITest_UserSettings.m to the new one and make sure to reset `COMPUTERID` in CITest.m.

Basic Operation

The main steps in running an experiment are 1) opening the GUI, 2) setting information about the subject and experimental session, 3) choosing the experiment type and mode (see definitions below), 4) setting stimulus parameters, 5) executing the experiment, and 6) assessing the results. Details of these procedures are outlined below.

For clarity, the *type* of experiment describes the form of stimulus and intended psychophysical measure. An example is the acquisition of perceptual threshold in response to an unmodulated pulse train. The *mode* of the experiment describes the way the stimulus is presented and the manner in which the data is measured. This will often dictate how the test subject interacts with the graphical user interface. An example is the 2-interval forced-choice method, in which the subject presses one of two buttons each time a pair of stimuli is played.

Lastly, note that the term *channel* is used below in a manner that is distinct from the term *electrode*. A channel refers to the set of intracochlear electrodes (and the extracochlear return electrode), their particular spatial configuration, and the relative amplitude of current that flows simultaneously through them. In this respect, different values of sigma and/or alpha (for partial tripolar and steered quadrupolar configurations) imply different channels.

<u>Launching CITest</u>

- Type `CITest` in the MATLAB command window to launch the GUI application

OR

- Open **CITest.m** in the MATLAB editor and run it using the green *">"* button at the top of the window, the *Run CITest* option located in the *Debug* menu, or the *<F5>* function key.

<u>SUBJECT INFORMATION Panel</u>

- In the *`SUBJECT ID`* field, enter the identification of the subject to be tested. Any string normally valid as a Windows folder name should be acceptable (i.e. spaces and certain special characters like "+" and "/" are not permitted).
    - If a directory for this subject does not already exist, a notification window will appear stating that a new directory will be created. The directory won't be created until after successful execution of an experiment run.
    - Invalid characters are automatically removed from a string entry, and if the final string is itself invalid for any reason, the entry will default to the last acceptable string. This type of safeguard is also used for other GUI string fields throughout CITest.
- Enter any positive integer in the *`SESSION NUMBER (SESN #)`* string field. This number will be used to name a subdirectory within the subject directory.

  - The motivation for this entry is that consecutive session numbers can be used to differentiate successive days of experiments for a given subject. No conflict will occur if the same session number is reused for a later day. [SMB: Perhaps a warning should be given if the specified SubjectID/Session## directory already exists?]

- The run number for the current session normally begins at 1. If desired, it can be set manually to another positive integer by changing the `RUN NUMBER (RUN)` field.
  - A warning will be given that manual changes can result in nonconsecutive and/or redundant run numbers for the current session. However, because file names are ultimately time stamped, there is little chance that data will be overwritten.
  - If the subject or session fields are changed, a reminder will be given to reset the run number to 1.

- Following an experiment run, results are saved in a file whose name includes information about the *type* of experiment and *mode* of subject interaction, as well as the electrode configuration, date and time of file saving, and run number. This file name, including the full directory path, is displayed at the top of CITest.
  - The directory naming convention is *Top-level/(Experiment Type)/SubjectID/Session##/*, using a leading zero for single-digit session numbers. The experiment directory level is optional and is set up in the user settings mfile. [SMB: It should be easy to make the file and directory naming specific to each user. Let me know.]
  - As an example, the file "THR-2IFC_20131128-1720_SQP050_EL08_Run02.mat" corresponds to a threshold experiment using the 2-interval forced-choice method and stimulating at electrode 8 in the steered quadrupolar mode with a sigma value of 0.50. The run number is 2 and the file was saved on November 28th, 2013 at 5:20 PM local time.
  - Changes to subject id, session or run number, experiment type or runtime mode, and certain stimulus parameters will cause the displayed file name to update.
  - Until the file is actually created, the displayed time stamp is "0000".

- The top-level data directory containing the subject folder (and optionally a folder corresponding to experiment type) is originally specified in **CITest_UserSettings.m**. It can be changed by pressing the small `[D]` button at the top-right of the panel.

<u>EXPERIMENT Panel</u>

- All available experiment types are listed in this panel. Choose one from the pull-down menu labelled *MAIN TYPE*.

- The *SUB TYPE* pull-down menu provides additional experimental options. This menu will generally be different for each main experiment type. Many sub types will specify the type of puse shape used for stimulation (see next). The choice of sub type could also determine what stimulus parameters are available in the EXPERIMENT PARAMETERS panel.
- Currently, "Standard Pulse Trains" is the first and default entry in the *SUB TYPE* menu for all experiments, which sets up stimuli consisting of biphasic pulses with no interphase gap. "Standard Single Pulse" (which allows better temporal specification for phase duration than pulse trains) is an option for all experiments except Psychophysical Tuning Curves. Future sub types might include pulse trains with a non-zero gap, or pseudo-monophasic pulse shapes. Note that these pulse shape attributes require different BEDCS definition files for implementation, but the experiment can otherwise be run using the exact same stimulus parameters and GUIs, which is the primary reason the experiment sub type was created (beginning with version 1.2). See the Pulse Modification section in STIMULUS PANEL.
- Changing the experiment main type (and, in some cases, sub type) automatically causes the EXPERIMENT PARAMETERS panel, located directly beneath, to populate with the appropriate UI controls.
- *START*: Once all the experiment information and stimulus parameters are set up, pressing this button will launch the next experimental run by opening up the subject interaction / runtime GUI.
- *PAUSE (||)*: During a run, pressing this button will temporarily halt execution. Via a popup window, the controller will be asked to continue or terminate the run. If the latter choice is made, the current trial may have to finish before the run is finally stopped (e.g. during a forced-choice procedure).

  - The keyboard shortcut for the pause button is *<P>*. Shortcuts are useful for experiments in which the test subject has control of the computer mouse.
- Right-clicking on the *START* button, EXPERIMENT panel, and SUBJECT INFORMATON panel brings up a context menu. The menu currently has two choices:

  - *Run In Demo Mode*: This toggles the demonstration mode on and off. See INSTALLATION for more information.
  - *Reset CITest*: This forces the CITest window to reset, which is useful if the runtime GUI or Results Window are behaving errantly, or if the *START* button is inactive. The CITest user settings and other start up procedures will not be re-executed, however.

- For really bad CITest hang-ups, type `delete(hctrl)` in the MATLAB command window, then call computer support in Seattle. (No, not Microsoft, the other guys.)

STIMULUS PARAMETERS Panel

- This panel provides control over stimulus parameters that are common among the various experiment types. The parameters relate to the *within trial* stimulus presentation of a single electrode channel or, in the case of dual-channel stimulation such as for forward masking paradigms, the primary channel. The presentation of multiple channels *across trials* can also be set up in this panel.
- In this and other panels, some of the UI controls will display a "tool tip" if the computer mouse is hovered over the control icon without clicking. The tool tip will contain a brief description about the stimulus parameter or additional information about the data displayed.
- *ACTIVE ELECTRODE*: This pull-down menu allows selection of a single active electrode for stimulation. If more than one electrode is desired for an experiment that allows multiple channels (see note above on channels versus electrodes), the *CHANNEL SELECT* menu should be used.
  - If multiple electrodes have been defined, the "tool tip" revealed by hovering the mouse over the pull-down menu icon will show the list of electrodes.
- *CONFIGURATION*: The choices of electrode configuration available with this pull-down menu are partial tripolar (pTP), bipolar (BP), and steered quadrupolar (sQP). The settings for *SIGMA* and *ALPHA* specify additional aspects of stimulation. Some further information:
  - sQP is like pTP, but it has two central "active" electrodes rather than one. For the purpose of naming channels numerically, the active electrode is the more basal of the two central electrodes, following the convention of Srinivasan et al (2012). As with pTP, the parameter σ (sigma) determines the amount of current flowing to the two electrodes on either side of these central electrodes, with the remainder flowing to the distant return electrode. A σ of 1 corresponds to the most spatially restricted pattern of current flow.
  - The sQP parameter α (alpha) scales the amount of current flowing from the basal active channel, with the balance, α – 1, flowing from the apical active channel. α = 1 corresponds to the total amount of active current flowing through the basal channel and α = 0 corresponds to all current flowing through the apical channel. Alpha does not affect the proportion of return current flowing through the flanking electrodes.

  - pTP can in theory also be steered with the α parameter, as in Wu and Luo (2013). However, because the flanking electrode currents are affected by both the α and σ parameters, current steering and spatial selectivity cannot be independently controlled. [For this reason, current steering with the pTP mode is disabled in CITest. *ALPHA* will be automatically set to 0.5, which for pTP is the "center" steering value.]
  - Monopolar (MP) can be specified with either the pTP or sQP modes by setting sigma to 0.0. Starting in version 1.20, a separate BEDCS definition will be used for MP, increasing the number of allowable active electrodes.
  - For the BP configuration, the active electrode is defined as the most basal of the electrode pair. The sigma parameter used for pTP and sQP becomes the electrode separation parameter, with 0 indicating a separation of one electrode. Thus, by convention, "BP+0" means that the return electrode is the one immediately adjacent to the active electrode in the apical direction.

- *SIGMA / BP SEP.*: For pTP and sQP modes, this text field is used to specify the current compensation parameter sigma, defined as the proportion of current from the active electrode(s) that flows to the flanking electrodes. For BP, the entry must be a nonnegative integer and corresponds to the number of electrodes between the active and return electrodes that form the bipolar pair. A value of 0 gives the standard bipolar mode using adjacent electrodes, a value of 1 gives the "BP+1" mode, etc.
  - Sigma (or BP separation) is a fixed value. If multiple channels have been defined using the *CHANNEL SELECT* menu, sigma will be the same for all channels.
- *ALPHA*: For pTP and sQP modes, this text field is used to specify the steering parameter alpha. See *CONFIGURATION* for more information.
  - One way to specify multiple channels for consecutive stimulation is to define a vector of alphas in *CHANNEL SELECT*. If alpha is not scalar, the text field will display the word "range" and its tool tip will reveal the number and range of alpha values (e.g. "5 values from 0.0 to 1.0").
- *PHASE DURATION*: This field specifies the duration in microseconds of the primary phase of each current pulse delivered by the stimulating channel. For the most typical type of stimulation, biphasic pulse trains, this parameter will specify the pulse width of the cathodic and anodic phases of each pulse. However, it could also be used to specify the baseline phase duration for stimuli using temporally modulated pulse widths or asymmetric pulses.
  - The tool tip for this text field will show the time base for the experiment, which should be accurate to a small fraction of a microsecond. The time base is the smallest

possible time interval for which elements of the pulse train, like phase duration, can be defined. It is a specific parameter within the BEDCS experiment files.

- For single pulses or other simple stimuli, the time base can be as small as 44/49 = 0.898 µs. For pulse train stimulation that requires the BEDCS function DSP_PERIODIC, the time base must be a multiple of 44/49*12 = 10.776 µs.
- Phase duration is automatically rounded to the closest integer multiple of the time base, after accounting for interphase gap.

- `PULSE RATE`: This is the number of pulses, biphasic or otherwise, that would occur if the pulse train duration were 1 second.
  - The value will update whenever `PULSE TRAIN DURATION` is changed, and vice-versa. Entered `PULSE RATE` values will automatically adjust such that an integer number of zero-amplitude time frames occurs between pulses. Thus, the pulse rate cannot be specified to arbitrary precision.
  - The tool tip displays the total number of zero-amplitude time frames between pulses. Note that `PULSE TRAIN DURATION` is precisely related to the number of inter-pulse zeroes by the following formula:

    #pulses x (#pulse frames + #zero frames) x (time base)

    For biphasic pulses with no interphase gap, the number of pulse frames is twice the `PHASE DURATION` value divided by the time base.
- `PULSE TRAIN DURATION`: This specifies the duration of the pulse train, in milliseconds. It is defined as the time from the first pulse to the end of the last inter-pulse interval (i.e. it includes the entire duty cycle for an integer number of pulses).
  - This value is also automatically rounded to the closest integer multiple of the time base.
  - The tool tip shows the total number of pulses, which depends on the `PULSE RATE`.
- `PULSE MODIFICATION (PULSE MOD >>)`: Pressing this button opens us a new GUI window containing additional information about the stimulus pulses, especially the shape of individual pulse waveforms. The pulse-related parameters available in this menu will depend on the chosen sub mode of the current experiment (see EXPERIMENT PANEL).
  - The default pulse shape is called “Simple Biphasic” and can be accessed by choosing the “Standard Pulse Train” experiment sub type. The two parameters for this mode are Initial Polarity and Interphase Gap. At present, neither parameter can be changed.

  - Some experiments are compatible with the “Simple Biphasic 1-Pulse” mode, which is suitable when only single pulses are needed. BEDCS experiment definitions for single pulse stimuli don’t use the DSP_PERIODIC function, so the time base duration can be shorter, allowing for more precise settings for parameters like phase duration and pulse rate. As with the standard pulse trains, currently none of the single-pulse parameters can be changed.
  - In a future release, expect to see new experiment types and subtypes that use more complex and modifiable pulse shapes, such as pseudo-monophasic pulses.

- `CHANNEL SELECTION (CH. SELECT >>)`: Pressing this button opens up a new GUI window, allowing the user to define multiple channels for sequential stimulation, set electrode-specific compliance limits, and do other things related to stimulus channels. This entry deserves its own subheading, which you’ll find below.
- `BEDCS VISIBILITY ([B])`: The BEDCS application is usually invisible during normal operation, because the new version otherwise crashes during stimulation. However, you can press this toggle button to view the BEDCS application window. In demonstration mode, which works without CI stimulation, this feature allows you to watch the BEDCS parameters change during a run. In normal mode, please close the BEDCS window before launching a run.
  - The text on the button turns green when the window is open, and is black otherwise.
- `RESULTS WINDOW ([R])`: This toggle button selects whether the Results Window GUI figure will be visible and active during an experiment.
  - During a run (i.e. when the subject interaction GUI is open), toggling will hide or display the Results Window. If the window is displayed, the experiment’s usual procedure for displaying ongoing results will be operational. If the window is not visible, the results may or may not be updated, depending on how the experiment was programmed. Thus, if the window is toggled back on later, there may or may not be gaps in the displayed results. (Either way, the data will be saved to file.)
  - Between runs, toggling off will close the Results Window, but toggling on will not reopen it until the next run is launched.
  - The reason for toggling the Results Window off is that there can be a significant processing overhead to displaying data during a run. Keeping the window closed can also prevent the test subject from inadvertently viewing results, especially in a one-monitor set up.
  - The keyboard shortcut is <*R*>.
  - The text on the button turns green when the window is open, and is black otherwise.

CHANNEL SELECTION Menu

- This menu, contained within a separate GUI window, is used to view and set a number of electrode properties and stimulus parameters, and to facilitate the creation of single or multiple electrode channels. For setting up a primary stimulation channel, the menu is accessed from the main CITest window by pressing the *CHANNEL SELECTION* button. Experiments requiring a secondary channel will use a separate button in the EXPERIMENT PARAMETERS panel for menu access.
- The Channel Selection window remains open until either the *SAVE* button, which retains all edits made within the menu, or the *CANCEL* button, which causes all changes to be ignored, has been pressed. While open, the main CITest GUI will be nonfunctional.
- The *LOAD* button is used to populate the menu elements with information that was used in a previous experiment run. The *IMPEDANCE (IMPED)* button can be used to run a fast electrode impedance routine and fill the impedance text fields with the results. [Neither of these buttons is currently operational.]
- Following is a list of the electrode properties and stimulation settings available in the Channel Selection menu:

  - *ON/OFF*: These radio buttons turn on and off the particular electrodes to be used for the next experiment. The electrode numbers are labeled underneath the radio buttons. At least one electrode must always be available. Electrodes that cannot be defined because of the current configuration (e.g. active electrodes 1 and 16 can't be used with tripolar) are automatically disabled. Note that the configuration itself is set in the main GUI. It is up to the user to avoid turning on a "bad" electrode.
  - *ELECTRODE*: This row is just a list of all electrodes for the implant, each of which corresponds to one column of editable UI text fields.
  - *M-ELEC / 1-ELEC*: When this button reads *M-ELEC*, pressing it turns on the ON/OFF radio button for every valid electrode. When the button reads *1-ELEC*, pressing it turns off all electrodes except electrode 8, making this a quick way to uncheck many radio buttons at once.
  - *THRESHOLD (THR)*: Each text field in this row can be used to display a known threshold current level for an active electrode, corresponding to a particular set of stimulus parameters (see *STIM PARAMETERS* below). For this and other menu text fields, a '--' symbol means that no data for that electrode is available. Once a set of thresholds have been obtained, the user can manually type in the values, in microamps, for the tested electrode. [SMB: In a future release, the threshold experiment will be able to push data into the Channel Select menu automatically.] Along with *MCL* and *COMPLIANCE LIMIT*, entries for threshold can be used to

specify relative current levels for stimulation (see the Current Level entry under the EXPERIMENT PARAMETERS heading).

- *MCL*: These text fields work the same way as for threshold, but the values reflect maximum comfort levels. Note that for both types of perceptual data, it is assumed that the entry relates to stimulation that is "centered" on the active electrode (i.e. in the absence of current steering).
- *IMPEDANCE*: These text fields are used for entering measured impedance values, in units of kOhm. The impedance is assumed to have been measured as a maximum peak value in response to a low level current pulse. Each impedance, in turn, determines the *COMPLIANCE LIMIT* for the corresponding electrode, using an algorithm that corrects for capacitive charging at the electrode-tissue interface. [SMB: In a future release, there will be a way to quickly measure impedances and automatically fill the UI fields.]
- *COMPLIANCE LIMIT (COMPL. LIMIT)*: These values cannot be altered manually, but are calculated directly from the electrode impedance. By default, the compliance limit for each electrode is 333 µA, which is much smaller than typical values.
- *STIM PARAMETERS*: This informational section of the menu displays the present set of stimulus parameters associated with the channels being selected. (For the primary channel, these are the parameters set in the STIMULUS PARAMETERS panel.) The threshold and MCL values listed in the menu apply as long as the stimulus parameters remain unchanged (see below).
- *ALPHA RANGE*: This popup menu can be used to specify multiple values for alpha, or a single value. Choices include the full range possible (0 to 1) or various preset subsets. During an experiment, the specified alpha value or values will be applied to every active electrode that has been turned ON. Thus, the total number of channels defined for an experiment will be (# of active electrodes) *x* (# of alphas). Note that even if only one active electrode is included for stimulation, multiple alpha values will define a multi-channel series.
- *ALPHA VALUE*: If a range of alpha values was chosen in the *ALPHA RANGE* popup menu, this text field specifies the step size from the start to end of the range, inclusive. The entered step size is automatically adjusted to assure an integer number of alphas. If a single alpha value was chosen in the popup menu, the text field instead specifies the value of alpha itself. For primary channels, this method of choosing a single-valued alpha is redundant with the *ALPHA* text field in the main GUI.

- After all desired changes have been made, pressing the `SAVE` button will close the window. The new electrode/channel settings will then apply to subsequent experiment runs. Pressing `CANCEL` will also close the window, but the settings will revert to their status prior to opening the Channel Selection menu.
- Changing certain settings in the main CITest window will reset the threshold and MCL information in the Channel Selection menu. This includes changes to the subject id or changes to configuration mode, sigma, phase duration, pulse train duration, or pulse rate. The reason for the reset is because threshold and MCL are specific to the subject being tested and the stimuli being presented.

EXPERIMENT PARAMETERS Panel

- The actual name of this panel changes to reflect the chosen experiment (e.g. THRESHOLD PARAMETERS). It contains all stimulus parameters that are specific to the experiment. More general parameters are defined in the STIMULUS PARAMETERS panel.
- More detailed information about experiment-specific parameters will be covered in the next section, Experiment Types.

Setting Current Levels

- One type of stimulus parameter set in the EXPERIMENT PARAMETERS panel that is common to all experiments is current level, usually in the form of a fixed level (as for gap detection experiments), or as starting and maximum levels (as for threshold experiments). For obvious reasons, current level is a very important parameter, so its handling by CITest will be elaborated here.
  - When the `START` button is pressed, a check for safe current levels is automatically made. Levels are checked against a sequence of criteria in the following order: 1) Absolute current limit for the device, which is around 2 mA; 2) Compliance limit, based on impedances set in the Channel Selection menu; 3) 10% above the maximum comfort level, if available for the active electrode (for sQP, alpha is used to weight each center electrode's contribution); 4) Typical maximum level for the configuration, which can be adjusted if necessary in the file **CITest_UserSettings.m**. Currents that exceed criterion levels 1 or 2 will cause the run to terminate and a warning message will be flashed to the user. Currents that exceed criterion levels 3 and 4 will flash a warning and the user be asked whether to continue. In either case, recheck all stimulus parameters and proceed cautiously!!
  - Current levels can be set relative to threshold, MCL, or compliance limits, or as a percentage of the dynamic range. CITest uses a special syntax to interpret text written

in the UI element that handles the level setting. The syntax will be auto-corrected as long as certain keywords are entered.

- The general syntax for writing relative levels is: *level operator value unit*. The keywords for base level are *thr*, *mcl*, *pdr*, and *cmp*. The keywords for operator are *+*, *-*, and *%*. The keywords for unit are *ua* and *db*. Value is a number expressed in the indicated unit.
- For levels relative to threshold or MCL, µA is the default unit. *%* refers to a linear percentage of the base value.

  examples*:* *thr + 10 ua,* *thr + 20%,* *mcl – 20 db*
- Percent dynamic range by default is interpreted on a *linear* scale between threshold and MCL. However, it can also be specified on a decibel scale, in which case the auto-syntax will read '*% dB*'.

  examples: *pdr + 10% ua* (10% of the dyn. range above thresh. on a linear scale)
  pdr + 40% db (40% of the dyn. range above thresh. on a dB scale)
- µA is also the default unit for levels expressed relative to the compliance limit. *%* is interpreted on a linear scale.

  examples*:* *cmp + 0 ua,* *cmp - 20%,* *cmp – 10 dB*
- Entries that attempt to invoke a base level that has not been defined in the Channel Selection menu will generate a warning when the `START` button is pressed.
- Levels can be expressed in dB (with respect to 1 uA) without reference to a base level. These are automatically translated in the UI field to a value in uA.

  examples: *42 dB* (automatically changed to 125.9 uA)

- At the conclusion of a run, data is either automatically saved to the file indicated at the top of CITest or the user is asked whether to save the data, according to the internal preferences of the experiment. Some type of post-processing and display of the data may also be automatically performed.

RESULTS VIEW Window

- The Results View window is where results of an experiment are presented during and after the run. It is a separate GUI from the main CITest GUI that is automatically opened when CITest starts. (Some post-run analyses may also display data in one or more additional windows.)
- The window consists of two panels, accessible from the radio buttons at the top-left of the figure. *CURRENT* displays an axes to which data is plotted during an experiment run.

SUMMARY displays a textual overview of the last run or related series of runs. The Summary panel is presently non-functioning, but this is where, for example, thresholds for all active electrodes would be listed following a channel sweep or series of 2-IFC runs.

- This window opens, if it is not already open and if the `[R]` toggle button is in the ON state, at the beginning of a run. It remains open at the end of the run to allow the user to view the results.
- Pressing the *[R]* button (in the STIMULUS PARAMETERS panel) to the OFF state will make the Results View window invisible, and the display axes will not be filled with data during the course of the next run. If *[R]* is then toggled ON after the run, the window will open but the display axes will remain unfilled (until the next run).
- The data displayed in either panel is dependent on both the experiment type and runtime mode. Thus, data generated during a threshold experiment run in channel sweep mode is presented differently than threshold data generated with a forced-choice paradigm.
- Closing the Results Window by pressing the "X" at the top-right corner, or by toggling if off using the CITest button, causes the window to be invisible and doesn't actually delete it. The MATLAB figure handle can always be accessed from the command line with `getappdata(hctrl,'hresults')` and the axes handle with `findobj (hresults,'tag',’axes_results’).`
- Pressing the *ANALYSIS* button will pull up a context menu from which scripts or functions can be run. Currently, the default option for all experiments will be to repeat the regular post-run CITest analysis. Any other routines have to be declared in CITEST_USERSETTINGS.M using the 'handles.customanalysis' setting specific to the experiment type. The rouine has to be a conforming function (ask Steve for details), but ultimate it can be based on the type of script one might run from the command line to process data in the CITest 'runResults' format.
- The *PUSH* button is currently not functional. In a future release, threshold, MCL, and loudness balance levels will all be data types that can be "pushed" to the Channel Selector menu for autofilling the associated UI fields. The idea is that this will facilitate the execution of experiments (like tuning curves) which rely on relative current levels.

Experiment Types

"Experiment type" reflects the kind of measurement being made. Each main experiment type (currently Threshold, MCL, and PTC) will have a different set of subtypes available, though "Standard Pulse Train" and "Standard Single Pulse" will usually be options. Both the main and sub types are chosen from pull down menus in the EXPERIMENT panel. Whenever an experiment type is chosen, the EXPERIMENT PARAMETERS panel automatically changes to display the appropriate UI fields for choosing experiment-specific stimulus and runtime parameters.

The following subsections will describe the options available in the EXPERIMENT PARAMETERS panel that are specific to the *kind* of data being collected. Many of the UI fields in the panel, however, depend on *how* the data are collected, which is referred to as “runtime mode” in CITest. These features are described in the next section, Runtime Modes.

Threshold and MCL

- Threshold and MCL experiments are handled with the exact same parameters and user interface. The only difference is in how the resulting file names are labelled.
- *`Starting Level (uA)`*: Set this to a valid numerical current level in uA or reference level (see Setting Current Levels) to indicate the starting current level for each channel. Proper choice of this parameter can save time.
- *`Maximum Level (uA)`*: This works the same as Starting Level, and helps prevent the runtime GUI from setting a current level too high.
- *`Message Log`*: This text display area isn’t currently being used. [SMB: In the future, this might be where messages are sent that are currently dumped to the command line. Or it might house a mini Tetris game, to pass away the time during tedious experiments.]

Psychophysical Tuning Curve

- For this experiment, the fields in the STIMULUS PARAMETERS panel correspond to the masking channel. Multiple maskers can be defined using the Channel Selection menu.
- *`Probe Fixed Level (uA)`*, etc : The active electrode, configuration parameter, level, and duration of the probe channel are set in the EXPERIMENT PARAMETERS panel (which will be labeled “PTC Parameters”). Only one probe electrode can be selected.
- The Channel Selection menu in this panel corresponds to the probe channel. Use this menu to make sure reference levels like threshold or MCL are set appropriately. (The compliance levels should automatically import from the Channel Selection menu of the STIMULUS PARAMETERS panel.)
- *`Masker Starting Level (uA)`*, etc : The range of masking levels is also set in the EXPERIMENT PARAMETERS panel.
- *`Masker-Probe Interval (ms)`*: This specifies the interval from the end of the masker train to the start of the probe train. The tip tool displays the total number of equivalent pulses that encompass this interval; it also displays an approximate of the

entire end-of-masker to start-of-probe duration by adding the last interpulse zero-amplitude interval that follows the last masker pulse (approximate because, for technical reasons, this does NOT include the duration of the 0-amplitude probe pulse that immediately follows the last masker pulse).

- Limitations of BEDCS require the masker and probe to have the same pulse rate. Electrode configuration also has to be the same, though the sigma value can be different. These UI fields are shaded grey and can't be adjusted. For sQP, the probe channel cannot be steered (i.e. alpha = 1).

Runtime Modes

A chosen experiment type and subtype is ultimately run with one of several possible "runtime" modes. A runtime mode determines the way stimuli are delivered and how the test subject interacts with the CITest program. The mode is chosen from the pulldown menu just below the EXPERIMENT PANEL label.

Manual Level

- All experiment types can be run in the *Manual Level* runtime mode. This mode will always be the first, and default, option in the Experiment Parameters pulldown menu.
- This mode was designed to test proper stimulus levels before launching the main experimental runtime mode (like channel sweep). In this way, threshold, maximum comfort, or loudness balanced current levels can be obtained manually using the exact stimulus (no matter how complex) that would normally be used for the main experiment.
- When the run is started, a popup menu is opened that either the experimenter or subject can interact with. Level can be adjusted in dB or linear increments using the *+* and **-** buttons. If the experiment has a "block" parameter , that can be changed using the *arrow* buttons. For threshold experiments, for example, the block parameter is channel number. All other parameters will be as listed in the main CITest window. When the desired level and parameter is set, press *PLAY* to deliver the stimulus.

Manual Parameter

- This mode was also designed to test stimuli before launching the intended experimental procedure. The single parameter that is manually adjusted in this mode is the same one that would be variable for the other runtime modes. For example, with gap detection, the pulse train gap would be the variable parameter. There may also be a block parameter.

- For some experiment types, other adjustable parameters might be available. These would be listed in the run mode pull-down menu as with names reflecting the changing parameter. For example, "Manual Mod. Freq." for an AM stimulus that can be changed in modulation frequency, even though normally the current level would be the adjustable parameter.
- The Manual Parameter option is currently not implemented for any experiment.

2-IFC and 3-IFC

- "IFC" is short for "interval forced choice", and the number indicates the number of intervals the subject is asked to choose among. Both modes use the same basic MATLAB code but different GUI windows.
- The subject initiates the start of the run, then, when prompted, presses the interval button they believe contained the sound.
- *Stimulus interval (ms)*: This specifies the time from the start of one interval to the start of the next, for each forced-choice trial. The value entered must be greater than the duration of the pulse train plus the internal BEDCS delay indicated in the user settings.
- *Init. Step Up/Down (dB)*: These fields control the starting step sizes from trial to trial, both upward and downward, of the roving parameter (e.g. current level for Threshold, masking level for PTC). (This parameter won't be labelled "dB" for roving parameters that aren't current; but for now that's all that these IFC modes handle.)
- *End Step Up/Down (dB)*: These control the ending step sizes for the roving parameter. The transition from start to end is set by the next set of UI fields.
- *# Rev Step/Stop/Avg*: The first entry indicates when the up/down step sizes should transistion from their initial to their end value (set with the UI text fields above this one), in terms of the number of up/down reversals obtained thus far in the run. The second entry indicates the number of reversals that causes the run to stop. The third entry indicates the number of reversals over which to average for the purpose of summarizing the data at the end of the run (this will not affect offline processing that the user might perform later).

Channel Sweep

- This runtime mode displays the exact same UI fields as the 2-IFC and 3-IFC modes, but some of them are applied differently or not used as all.

- `Forward | Backward | Both`: The direction of the sweep in channel (electrode number, alpha, or both) is set here. "Both" means that a backward sweep will immediately follow a forward sweep.
- `Stimulus interval (ms)`: This specifies the time interval between consecutive stimulus presentations.
- `Channel sweep interval (ms)`: This is the time elapsed between the first stimulus corresponding to one channel (electrode/alpha combination) and the first stimulus of the next channel, determining the rate at which channel number increases in value for a forward sweep or decreases in value for a backward sweep. It is forced to be an integer multiple of `Stimulus interval`.

Advanced Operation

Many of the UI elements for the various experiment types used to set runtime control parameters are actually the same. `Stimulus Interval`, for example. These shared UI elements are managed by callback functions located within the mfile associated with the Threshold experiment type (**CITest_Exp_Threshold.m**).

Future Updates

Listed here are code changes currently being working on, as well as improvements that are planned but not yet started. Please report any bugs you encounter in the software and I'll try to fix them (or you can send me your own fixes) and make the update available to everyone as quickly as possible. Also feel free to request new experiment types or runtime modes that your research requires, or features that would improve the general usability of CITest.

In the works

- A new experiment type for loudness balancing. Or maybe a runtime mode for loudness balancing. The balance is level on this decision.
- If there is interest, I could implement current steering with the pTP configuration (as used in papers by Wu + Luo (JARO, 2013) and others). With regard to the previous entry, an alpha value of 0 or 1 with pTP at sigma = 1 is equivalent to a bipolar pair. In some ways, this is non-ideal for current steering because the function of the return current compensation, which is to narrow the electrical field, is undermined by the steering, which flattens the electrical field to become more bipolar at the extremes. For now, keeping the compensating and steering electrodes completely separate, as is done with the sQP configuration, seems preferable.

- Keep track of Channel Selector data during and between sessions. This will facilitate loading past settings (like threshold and MCL values) into the interface for a desired set of stimulus parameters (electrode configuration, phase duration, etc), so that these values don't have to be manually entered.
- Introduce subject profiles, to make it easier to keep track of past experimental runs conducted on each test subject.
- Fill in the Advanced Operation section of this documentation. This part is intended to describe more about the inner workings of the CITest software, to make it easier to fix errors or add experiments and features.

Glimmers of hope

- Electrode impedance measurement. The idea is to incorporate this into the Channel Selector interface, where the impedance fields can be directly filled by the measured values.
- If a need arises to more precisely set the timing between pulsed trains, I will work with Leo to try to optimize the MATLAB code and its interface with BEDCS, and see if there are any operating system settings that can be adjusted. It may be necessary to calibrate and keep track of processing delays for every particular experiment and runtime mode, as well as for different computers running the software.