CITest Documentation Version 1.13, Dec 05, 2013

# 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. Much of its functionality originates from an earlier application, called “CI\_Test”, created by Julie Bierer 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]

# New In This Version :: Current Experiment Types and Runtime Modes

Version 1.13 is the first version of the revamped CITest software made available to labs outside of the University of Washington. It can be run using any PC-based version of MATLAB from R2007b and onward. Currently, only the threshold and maximum comfort level experiment types can be executed, using manual, 2-interval forced choice, and tracking procedures.

At the time of future CITest updates, look in this section for descriptions of any software changes and related user instructions and tips.

# Installation

* 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 (e.g. CITest\_v01.13) 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.
* Copy the folder called “CITest\_BEDCS” to your computer at a convenient location, and optionally rename it. This folder contains BEDCS experiment definitions that are called from CITest. Possible locations include the same directory where CITest is stored, or the directory where other BEDCS experiment files are stored.
* 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 changing the user id.) This 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 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 your user setting to be called. 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 62). 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 is not running 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.

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.

Launching CITest

* 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.

SUBJECT INFORMATION Panel

* 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.

EXPERIMENT TYPE Panel

* All available experiment types are listed in this panel. Choose one by pressing the corresponding radio button.
* Changing the experiment 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.
* The *START* button and the SUBJECT INFORMATON panel provide access to a context menu by right-clicking on the UI element. 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 a single channel of stimulation or, in the case of dual-channel stimulation such as for forward masking paradigms, the primary channel.
* 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.
  + For the BP configuration, the active electrode is defined as the most basal of the electrode pair. [Bipolar is currently not implemented for the THRESHOLD experiment type, but should be soon.]
* *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. [For the “CITest\_pulsetrain” series of BEDCS experiment files, which are the only .bExp files that CITest currently uses, pulses are biphasic with the leading phase cathodic on the active electrode(s).]
  + 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.
  + Note that inter-phase interval is currently assumed to be zero. [If a future experiment type requires a non-zero interval, the inter-dependence of pulse train duration and pulse rate with phase duration will have to be altered.]
* *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 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, the number of pulse frames is twice the *PHASE DURATION* value divided by the time base.

* *SINGLE PULSE (1-PULSE)*: This check-box option is intended for experiments that permit single pulses, as opposed to trains of pulses. BEDCS experiment definitions for single pulse stimuli don’t use the DSP\_PERIODIC function in BEDCS, so the time base duration can be shorter, allowing for more precise settings for parameters like phase duration and pulse rate.
  + [SMB: The single pulse option is not currently available, as there are no BEDCS files that support it. The framework to incorporate it is in place, so let me know if you need this option.]
* *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 menu takes a few seconds to open as it populates the UI elements with the most recent settings. The 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.
* One type of stimulus parameter set in this 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.
* 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 WINDOW

* 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.
* The data displayed 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 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 getappdata (hctrl,'hview').

# Experiment Types

The Manual Runtime Mode

* All experiment types can be run in the MANUAL runtime mode. That is, this mode will always be an option in the Experiment Parameters pulldown menu.
* This mode was designed for the user to test any stimulus before launching the main 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 (aside from the exception described next).
* A variant of MANUAL mode (called MANUAL LEVEL) will be available for some experiment types in which current level is the adjustable parameter, even for experiments where current would otherwise be fixed. In this way, thresholds, maximum comfort levels, and other perceptual data can be obtained manually using the exact stimulus (no matter how complex) that would normally be used for the main experiment.

Threshold and MCL

* *# Rev Step/Stop/Avg*: For the 2IFC mode, 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 parameter 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).
* *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

* Stay tuned.

Gap Detection

* More to come.

# Advanced Operation

Look here for periodic updates about the inner workings of the CITest software.

# 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 the GUI.

In the works

* New experiment types for gap detection and psychophysical tuning curves.
* Fully implement the bipolar electrode configuration. The sigma parameter (internally called ‘config2’) can be used to indicate the separation of electrodes, with + or – specifying that the return electrode is apically or basally oriented from the active electrode. Alpha will be ignored, as there will be no current steering.
* 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.to
* Add a “give feedback: yes/no” option to Threshold :: 2IFC experiments. Right now, feedback is always on.
* 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.
* The 2-IFC buttons don’t revert to their neutral form when pressed. This is a JAVA issue that I hope to fix soon.

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.
* Create blank, or skeleton, GUI m-files to guide the design of new experiment and run-time modules, and make it easy to add them to the main CITest GUI. This way, external users can create new experiments (with the option of making them available to others).
* 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.