CCi-MOBILE Research Interface
for Cochlear Implant and Hearing-Aid Research
Release Notes: Jan-22-2019

These release notes address the following from version 2.2c:

- Realtime vs. offline processing
- Operating specifications – non-integer number of pulses per frame
- Data loss
- Other possible solutions for non-integer pulses per frame
- User-specified MAP and signal processing parameters
- Data communication

Question: Is there a difference between real-time and offline processing?
ANSWER: Yes. Differences between real-time and offline include: pre-emphasis step, user-specific parameter adjustment, communication to the board, and a few minor function calls/for loops. The real-time implementation was specifically designed to minimize the latency in data communication and signal processing. The offline implementation was predominately adopted from Nucleus MATLAB Toolbox (Cochlear Corp.)

- For engineers, compare the functions (CCi-MOBILE\MATLAB\AudioFileProcessor) 'ACE_Process.m' and (CCi-MOBIE\MATLAB\Realtime) 'ACE_Processing_Realtime.m'

Question: What is the operating specification: 'non-integer number of pulses per frame'?
ANSWER: Non-integer pulses per frame will result in a selecting a percentage of a single sample which is not permissible (i.e. 4 samples verses 4.348 samples). The number of pulses per frame per stimulation cycle (RF cycle) depends on the stimulation rate (Hz) and the number of selected electrodes ('n-maxima') as demonstrated in equation [1]. The number of samples used in the overlap-and-add method (i.e. the variable, 'block_shift') must be an integer number of samples as demonstrated in equation [2]. To account for this, a change in the stimulation rate will be made to create an integer number of pulses.

$$Pulses_{total\ for\ 1\ RF\ cycle} = \left[\frac{(Frame_{(seconds)}*Fs)}{\left(\frac{Fs}{Stimulation\ Rate}\right)}\right] * [('n - maxima')] \qquad [1]$$

$$Block\ shift_{(samples)} = \left[\frac{Fs}{(Stimulation\ Rate)}\right] \qquad [2]$$

- For engineers, see the function (CCi-MOBILE\MATLAB\CommonFunctions\...) 'initialize_ACE_integer_ppf.m' lines 29-48 and lines 187-206 (ver 2.2c) for real-time implementation

Question: Is CCi-MOBILE losing data and/or dropping samples?
ANSWER: It is dependent on the real-time and/or offline implementation.

- For offline ACE/CIS implementation, there is not a loss in data. The remaining non-integer data bytes (RF cycles) are processed with the subsequent frame and the remaining byte-size shift is applied throughout the entire length of the offline speech token to provide stimulation without data loss. Because the signal is processing offline, the buffer function (overlap-and-add) will automatically zero pad the data to ensure all frames can be processed. **NOTE:** Adjustments to the user-specified parameters may ensure no data is lost.
  - For engineers, see the function (CCi-MOBILE\MATLAB\AudioFileProcessor\...) 'Stream.m' lines 38-46 and lines 54-62 (ver 2.2c)
- For real-time ACE/CIS implementation, there is a loss in data. The numerical amount of data loss is in the form of samples. The numerical amount of samples per stimulation cycle (RF cycle) is dependent on the user-specified stimulation rate. The maximum number of dropped samples is equivalent to: (sampling frequency) / (stimulation rate)[1]. **NOTE:** Adjustments to the user-specified parameters may ensure no data is lost.
  - For engineers, see the function (CCi-MOBILE\MATLAB\Realtime\...) 'ACE_Processing_Realtime.m' line 7 (ver 2.2c), variable output 'z' are the dropped samples.
  - **NOTE:** While there is a loss in a small amount of data for the real-time implementation of ACE/CIS, the subject performance (N=8) with their clinical processor against the real-time implementation of ACE/CIS

[1] NOTE: A non-integer value from this calculation will be rounded toward positive infinity (i.e. the MATLAB function, 'ceil')

through CCi-MOBILE did not result in a significant (p>0.05) difference in sentence intelligibility (results shared at CIAP 2017).

Question: Can we simply use up sampling/down sampling to generate exact stimulation rates?
ANSWER. You could pending change to software and hardware routines. This is how we believe the clinical processors from Cochlear Corp ensure exact stimulation rates. The software and hardware routines currently are hard-wired for sampling rates of 16,000 samples/sec, a frame size of 8ms, and an interphase gap (IPG) of 8 us. To support the proposed idea in real-time, there is a possibility of a latency. To support the proposed idea in general, modifications to the communicative hardware routines would need to be updated.

Question: Can we amend dropped samples from non-integer pulses per frame in the next frame (i.e. following MATLAB's 'buffer' command)?
ANSWER. You could, but this will raise additional problems. The CI Lab at UT-Dallas developed a solution (suggested by UWM) to amend the previously dropped samples from the prior 8ms stimulation cycle and continued to pass dropped variables to the subsequent 8ms cycle. Based on both a 5s and 10s run time for real-time simulated by our lab, minimal latencies were observed from comparing the current pipeline to the proposed pipeline (~5.2ms ± 0.012). This required a frame-by-frame change in the communication to the buffer to denote a change in the number of pulses per second within a stimulation cycle (RF cycle) due to an accumulation of sufficient dropped samples resulting in the need for an additional RF cycle periodically (depending on the overlap, i.e. fs/stim-rate). Additionally, the corresponding frequency components of varying frame-by-frame analysis when compared one-to-one to the current implementation resulted in a large (unquantified) change in the corresponding clinical levels and selected electrodes (if 'n-maxima' < total number of electrodes). This may induce a perceptual change that has not been tested on CI subjects.

Question: Is CCi-MOBILE adjusting my user-specific signal processing (or MAP) parameters?
ANSWER: Sometimes. Currently, the architecture of the user-specified parameter checking algorithm ensures minimum/maximum pulse widths are met for each 8ms frame. To compensate for parameters outside of the operating region (i.e. total stimulation rate > 14400 pulses per second, or pulse width < 25 Hz or > 400 Hz), a reduction of the user-specific stimulation rate is required to ensure an integer value of samples for each frame. This adjustment occurs during the initialization process.

- For engineers, see the function (CCi-MOBILE\MATLAB\CommonFunctions\...) 'initialize_ACE.m' lines 29-45 and lines 184-200 (ver 2.2c) for offline implementation
- For engineers, see the function (CCi-MOBILE\MATLAB\CommonFunctions\...) 'initialize_ACE_integer_ppf.m' lines 29-48 and lines 187-206 (ver 2.2c) for real-time implementation

RESPONSE: From our investigation of the software routines, the current state of the user-specific adjustment does NOT include a notification (verbosity) when an adjustment is made, and the algorithm, in fact, contains loops that do not update the variable outside of their function location. This can result in the user-specified stimulation rate to produce non-integer pulses per frame, total stimulation rate (Hz) outside of the operating specifications, and in some cases a different stimulation rate than the user specified.

- The CI Lab at UT-Dallas is developing the software release, ver 3.0 which will include a notification system to the user during adjustment, a GUI MAP parameter check (including signal processing parameters), priority parameters (hierarchy) for non-adjustment, and additional documentation/README support.

Question: Can I stimulate 116 bytes or 115 bytes per ear according to the UART buffer communication?
ANSWER: 116 bytes.

Question: Can I stimulate more than 116 bytes per ear due to unused space allocated in the buffer communication?
ANSWER: No. The development of the UART communication buffer specified particular amount of data within the first 258 bytes. This resulted in an asymmetrical organization of the buffer. Currently, the buffer includes reserved bytes (26 bytes). These reserved bytes are for a header when performing bimodal processing. To increase the amount of total information sent to the CI user, a hardware update would be needed as well as communicative routines from the software to the board. This would require updating all boards distributed in the field.

- The buffer structure is provided in a separate (.pdf) here.