Tuning Curve Client Quick Start Guide
6/18/04

Overview and Purpose

Tuning Curve Client (TCClient) is used to determine whether or not unit firings are correlated in time with some external event. TCClient allows the user to specify a set of units and Events that are of interest, then constructs peri-event histograms for each (unit,Event) pair. Each peri-event histogram bins the times of the unit firing relative to the time of the event (which is taken to be at time 0). Thus a peak in the peri-event histogram at a certain time relative to the event may mean that the unit is correlated with the event. TCClient can display these peri-event histograms, much like the PEC client. TCClient then goes one step further and uses the peri-event histograms to construct a tuning curve for each unit.

A tuning curve is a plot, made for each unit, which displays unit activity (or "Response") on the y axis versus events on the x axis. If a unit is correlated with a specific event ("tuned" to it), there will be a peak in the Response for that event on the tuning curve for that unit.

TCClient has expanded the definition of what constitutes an "Event". For TCClient, an Event is either a traditional Plexon external event (either individual TTL or strobed), or it can be the firing of a unit. A tuning curve that is based on a unit firing "Event" reveals cross-correlations between units.

TCClient can be both an online monitoring tool and an offline analysis tool, because it can either connect as an online client, or can play back a previously-recorded plx file. The peri-event histograms and the tuning curves are continually updated as data arrives.

More Detailed Information

Response Definitions

TCClient allows several options for defining the Response value that is plotted on the tuning curves. Fundamentally, a Response for a (unit,Event) pair is calculated by summing the bin contents within a "Response Window" in the peri-event histogram for that (unit,Event) pair, over some amount of time. TCClient allows the user to define the characteristics of the peri-event histograms (e.g. use 30 bins, from 100 mS before the Event to 200 mS after the Event), the width and position of the Response Window within the peri-event histogram (e.g. use bins 10-20 as the Response Window, which corresponds to a time range of 0-100 mS after the Event), and the amount of time to sum over (e.g. look back over the last 30 seconds of activity). The Response can be defined to be either:

- *Counts per Event* : the integrated counts in the Response Window divided by the number of Events that occurred within the time period, or
- *Counts per Event per Second* : the above, but divided by the time period

The user can also choose to use background-subtracted versions of either of the above Responses. When subtracting background, one of two methods can be used to define the background:

- *Peri-Event Activity for a Designated Event* : the user designates one of the defined Events as a special, background-determining event. The Response is calculated for this Background Event using the Response Window, same as is done for all other Events. This Background Response is

then subtracted from the normal Response calculated for each of the other Events. The same calculated Background Response value is subtracted from all other Responses.
- *Background Response Window* : the user defines another, independent Background Response Window that is used to calculate a Background Response for each peri-event histogram. This Background Response is subtracted from the actual Response calculated from the same peri-event histogram. Note that since the Background Response is calculated for each peri-event histogram, the amount of Background Response subtracted from each Response can be different.

Events and Units Definitions

A large part of configuring TCClient is to define the Events and Units of interest. TCClient will only pay attention to a subset of all the events and unit firings that it sees in the data stream. Defining this subset of Events and Units is done using the Events and Units views on the left side of the TCClient window (by default). These views provide an editable list of all the current Events and Units.

There are three methods to define new Events and Units: automatic, one-shot, and manual specification.
Automatic : This method is the easiest, but can only be used while data is flowing (TCClient has already been connected to a Data Source). In this mode, TCClient will monitor the data stream for all Events and Unit firings that occur, and automatically add them to the list. To activate this mode, check the 'Automatically Add' checkbox in the Events or Units view. Note that in this mode it is possible to overwhelm TCClient with large numbers of Events and Units, so the number of events or units that can be added in this mode is currently limited to 20. It is not possible to define Events that are unit firings using this method.

One-shot : This method is primarily intended for defining Events, though it also works for Units. In this mode, after a button is pressed, TCClient will automatically add the next Event or Unit that appears in the data stream. This is useful for defining external stimulus events that can be manually initiated. To use, press the 'New (Insert)' button in the Events or Units view (the toolbar button that looks like a dotted box), which brings up the 'Add New Event' or 'Add New Unit' dialogs. These dialogs have a 'Capture Next' button that initiates the one-shot definition sequence. For Events, you can control what type of Event will be added using the radio buttons. The capture sequence lasts for 10 seconds; if no Units or Events (of the requested type) occur within 10 seconds, the capture fails and the button must be pressed again to initiate another capture. When the capture succeeds, the dialog is populated with the information about the captured event (as if the user had specified it using the manual method below) but the Unit or Event is not added until the OK button is pressed.

Manual : To use, again press the 'New (Insert)' toolbar button. Adjust the controls in the dialog to designate the desired Unit or Event. Note that the 'Name' edit box is automatically populated with a suggested name for the Unit or Event, but the user can change the name at any time.

The Events and Units views can also be used to select the current Event and the current Unit by single-clicking the Event or Unit in the list. The selected Event and Unit affects which plots are shown in the TCClient views. Events and Units can be deleted (either singly or multiply) and re-ordered.

Events and Units can be inspected more closely and edited by double-clicking the Event or Unit in the list, which brings up a dialog. Only the names can be changed once a Unit or Event is defined.

.tcc files

Once you configure TCClient appropriately for your experiment setup, the configuration can be saved into a .tcc file, which can be loaded back into TCClient. Note that TCClient has an option where it can automatically load a .tcc file when it starts. To enable this, check the checkbox under 'Startup Behavior' in the Tools-Options dialog and select the .tcc file to load. Information saved to the .tcc file includes all Event and Unit definitions and all binning and display options.

Notes About the User Interface

The main views in TCClient are either in the docked or floating state. When TCClient comes up the default selection of views are all docked. Docked views can be resized by click-dragging the divider lines between them. Views can be moved by click-dragging on the title bar. By default, moving a view will result in the view docking at the final position. To leave the moved window in the floating state instead, hold down the Control key while click-dragging the view. Views can be closed, and can be opened again using the View menu.

By default, there are three columns of views in TCClient; the left column contains the Control, Events and Units views, and bar charts for Event and Unit activity. The middle column views contain grid views that show relatively crude renderings of all of the plots, and the right column views show a more detailed, magnified version of the selected plot. For the middle and right columns, the top views show the tuning curves (the grid containing all tuning curves, and the blow-up of the selected tuning curve), and the bottom views show the peri-event histograms (the grid view showing all Peri-Event histograms for the selected Event, and the blow-up of the peri-event histogram for the selected Event and Unit).

Most of the views in TCClient can be customized via the 'Properties' selection in a right-click context menu.

Usage

Start the Tuning Curve Client by double-clicking on TCClient.exe.

For the first-time configuration of TCClient, the following steps should be followed:

1) Select your options.
First, set some global options, such as what Response you would like to show on the tuning curves, and the update rates and look-back times. To do this, select Tools->Options, or click on the OPT… icon in the toolbar.

2) Select your binning parameters.
This sets up parameters having to do with histogram binning, like setting the time range and number of bins for the peri-event histograms, and setting the Response Window for calculating the tuning curves. Select View->Binning Options, or click on the icon in the toolbar. Hit Apply when you are happy with the values.

3) (optional) Start your data source.
The reason for doing this now is to take advantage of the Automatic and One-shot Event and Unit definition features of TCClient, which can only be used when data is flowing. From the Data Source menu, select either 'Connect' to connect as a client to a MAP data stream, or select

'Replay Existing File…' to play back an existing plx file. When using the 'Connect' feature, make sure that SortClient is running and the MAP system is already collecting data. Or, you can get connected by bringing up the Data Source View from the View->Data Source or Data Source->Show Data Source View menu items. If you don't start the data source here, do it as step 7.

4) Define your Events.
This is described in the 'Events and Units Definitions' section above

5) Define your units.
This is described in the 'Events and Units Definitions' section above

6) If desired, configure background subtraction mode.
To enable background subtraction mode (described above), bring up the Options dialog (Tools->Options…) and check the 'Subtract Background Response' checkbox. Then use the radio buttons to select which of the two background subtraction modes to use.
For 'Peri-Event Activity for a Designated Event' mode, an Event must now be designated as the background-determining event. From the Events view, double-click the desired background event. This will bring up the 'Edit Event' dialog. Check the 'Designate as Background Determining Event' checkbox. Note that the Background Event is now flagged with a '*BKG*' prefix in the Events view list.
For 'Background Response Window' mode, the Background Response Window must now be defined. Bring up the Binning Options view, and use the droplists in the 'Response Window for Tuning Curve Background' to define the Background Response Window. Note that Background Response Window is (optionally) displayed on the peri-event histograms.

You can now save a .tcc file that preserves the changes you have made.

How it Works

At the heart of TCClient are three large circular buffers, whose purpose it is to hold historical data, back in time. The most recent data is always at the 'head' of the circular buffer, and the oldest data is at the tail, where it will be overwritten. The position of the head pointer of each buffer is displayed in the 'Control' window as colored bars; the bars should advance as data is processed by TCClient. The three buffers are:
The 'Source' buffer : contains a history of the raw data coming in, condensed into a more space-efficient format.
The 'Histos' buffer : contains 'snapshots' of peri-event histograms for each Event occurrence.
The 'Response' buffer : contains a history of the instantaneous calculated Responses for each (unit,Event) pair.
The sizes of these buffers can be adjusted.

As data comes in, it is stored in the Source buffer. Whenever data arrives, TCClient looks at each unit firing or event, and decides one of three things:

a) the event or unit firing is not of interest, ignore it
b) the event or unit firing is an Event of interest – create 'active' peri-event histograms for it (one histogram for each unit).
c) the unit firing is of interest, bin it in the appropriate active (unit,Event) peri-event histograms

At any given time, there are a set of ‘active’ peri-event histograms, each one corresponding to an Event occurrence. Active peri-event histograms are those whose time window is still open – unit firings that arrive within the time window will get binned in the active peri-event histogram for that (unit,Event) pair. When the time window closes for an active peri-event histogram, the bin counts for that histogram are transferred into the Histos buffer. The Histos buffer contains snapshots of peri-event histograms; for each Event occurrence, there will be #units histograms in the Histos buffer.

Periodically (with a user-specified period, 2 seconds by default) TCClient will recalculate all of the Responses for the tuning curves. For the recalculation, TCClient will look back in time a user-specified number of seconds (30 by default) in the Histos buffer, and sum up the counts in the bins within the Response Window for all the peri-event histogram snapshots it finds within that time period. It then calculates the Response for each (unit,Event) pair (taking background-subtraction into account if relevant), and stores the calculated values the Responses buffer.

The Response buffer conceptually contains a time history of (unit,Event,response,errorbar) tuples. Currently, the tuning curve plots in TCClient always plot the very latest Response buffer entries for each (unit,Event) pair. However, in the future TCClient may display plots that show the time-evolution of Responses.

The Source buffer must be large enough to hold enough data so that TCClient doesn’t ‘fall behind’ when it is creating the peri-event histograms. Faster event rates and computers with slower CPUs should increase the size of this buffer.

The Histos buffer must be large enough to hold all relevant histograms for all (Event,unit) pairs for the entire recalculation interval (30 seconds by default). This is highly dependent on Event rate, but independent of unit firing rate. High event rates should increase the size of this buffer.

The Response buffer currently must only be large enough to hold the latest (unit,Event,response,errorbar) tuples for all unit and Events of interest. In the future, it may need to be larger to hold a history of these values, in which case the size would be determined by how much Response history would need to be displayed.