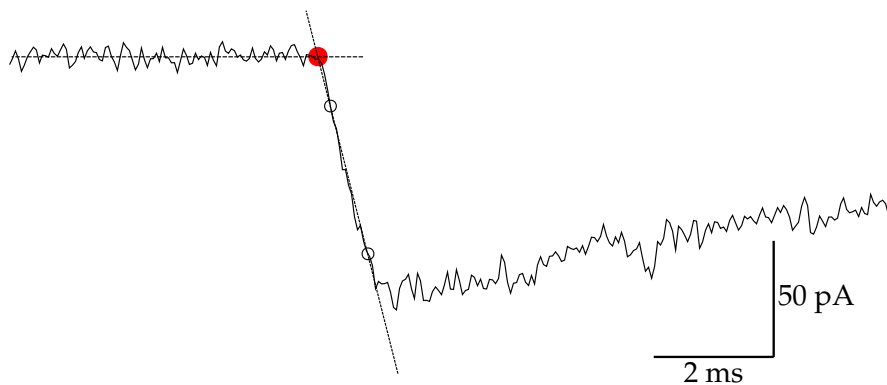# 3 Latency measurements

3.1 Measurement of synaptic delay

Stimfit is frequently used to measure the delay between a presynaptic signal and a post- *synaptic delay*
synaptic response. Classically, this synaptic delay or latency is defined as "the time inter-
val between peak of inward current through the presynaptic membrane and commence-
ment of inward current through the postsynaptic membrane" (Katz and Miledi, 1965).
Neglecting cable properties of neurons for a while, the maximal inward current during
an action potential is expected to flow at the time of maximal slope during the rising
phase (Jack et al., 1983), since

$$I_{\mathrm{m}} = I_{\mathrm{cap}} + I_{\mathrm{ionic}} = C_{\mathrm{m}}\frac{dV_{\mathrm{m}}}{dt} + I_{\mathrm{ionic}} = 0, \text{ and hence}$$

$$I_{\mathrm{ionic}} = -I_{\mathrm{cap}} = -C_{\mathrm{m}}\frac{dV_{\mathrm{m}}}{dt}$$

The commencement (sometimes called "foot") of the postsynaptic current can robustly *foot*
be estimated from the extrapolated intersection of the baseline with a line through the
two points of time when the current is 20 and 80% of the peak current (Jonas et al. (1993);
Bartos et al. (2001); Fig. 18).

Fig. 18: Foot of an EPSC (red circle), estimated from the extrapolated intersection of
the baseline with a line through the two points of time when the current is 20 and 80%
of the peak current (black open circles).

Although the method described above yields reliable results when both the pre-
and the postsynaptic whole-cell recording show little noise and few artefacts, it may
sometimes be favourable to use other estimates for the pre- and postsynaptic signals,
for example when extracellular stimulation was used or when there are a lot of failures
in the postsynaptic response. The following sections will explain how this is done in
practice.

3.2 Trace alignment

trace alignment It may sometimes be useful to align traces before measuring the latency, either for visu-
alisation purposes or to create an average without temporal jitter. Although an aligned
average can be created using a toolbar button, the recommended way to align traces is
to use the Python shell.

align_selected **align_selected(alignment, active=False)**
This function aligns the selected traces to a point that is determined by the user-supplied
function `alignment` and then shows the aligned traces in a new window. The `alignment`
function is applied to the active channel if `active=True` or to the inactive channel if
`active=False`. The alignment function has to return an index within a trace, and it
should adhere to the general form `index(active)`, where `active` is a boolean indi-
cating whether the active or the inactive channel should be used. The most common
alignment functions are built into the program:

maxrise_index **maxrise_index(active)**
returns the zero-based index of the maximal slope of rise in units of sampling points
(see Fig. 13), interpolated between adjacent sampling points, or a negative value upon
failure.

peak_index **peak_index(active)**
returns the zero-based index of the peak value in units of sampling points (see Fig. 13),
or a negative value upon failure. The return value may be interpolated if a moving
average is used for the peak calculation (see p. 6).

foot_index **foot_index(active)**
returns the zero-based index of the foot of an event, as described on p. 21 and in Fig.
18, or a negative value upon failure.

t50left_index **t50left_index(active)**
returns the zero-based index of the left half-maximal amplitude in units of sampling
points (see Fig. 13), or a negative value upon failure. The return value will be interpo-
lated between sampling points.

t50right_index **t50right_index(active)**
returns the zero-based index of the right half-maximal amplitude in units of sampling
points (see Fig. 13), or a negative value upon failure. The return value will be interpo-

lated between sampling points.

Listing 3 shows a function that can be used to align all traces within a file to the maximal slope of rise in the inactive channel:

```python
# import the stimfit core module:
import stf

def align_maxrise():
    """Aligns all traces to the maximal slope of rise
    of the inactive channel. Baseline and peak cursors
    have to be set appropriately before using this
    function.

    Return value:
    True upon success, False otherwise."""

    stf.select_all()

    # check whether there is an inactive channel at all:
    if ( stf.maxrise_index( False ) < 0 ):
        print "File not open, or no second channel; \
               aborting now"
        return False

    stf.align_selected( stf.maxrise_index, False )
    return True
```

Listing 3: `align_maxrise`

3.3 Setting the latency cursors

The latency cursors (plottes as dotted vertical blue lines) can either be set automatically   *latency cursors*
to some predefined points within a trace, or manually using the mouse buttons. The predefined points can be chosen from the menu: "Edit"→"Measure latency from..." and "Edit"→"Measure latency to...". The "beginning" of an event refers to the foot as explained above (Fig. 18). If "Manually" is selected, the left and right mouse buttons can be used to set the first and second latency cursors while the latency mode is activated. To switch to the latency mode, you can either click the corresponding button in the toolbar (Fig. 19) or press L.
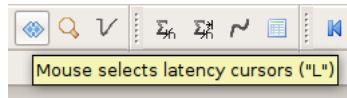
Fig. 19: Activate latency mode.

To confirm your latency cursor settings and measure latencies, you can either press $\boxed{\text{Enter}}$ or call measure() from the shell (see p. 15). The latency, i.e. the time interval between the first and the second latency cursor, will be shown in the results table as long as you activated this value (see p. 5). The latency will be indicated as a double-headed arrow connecting the two latency cursors (Fig. 20).
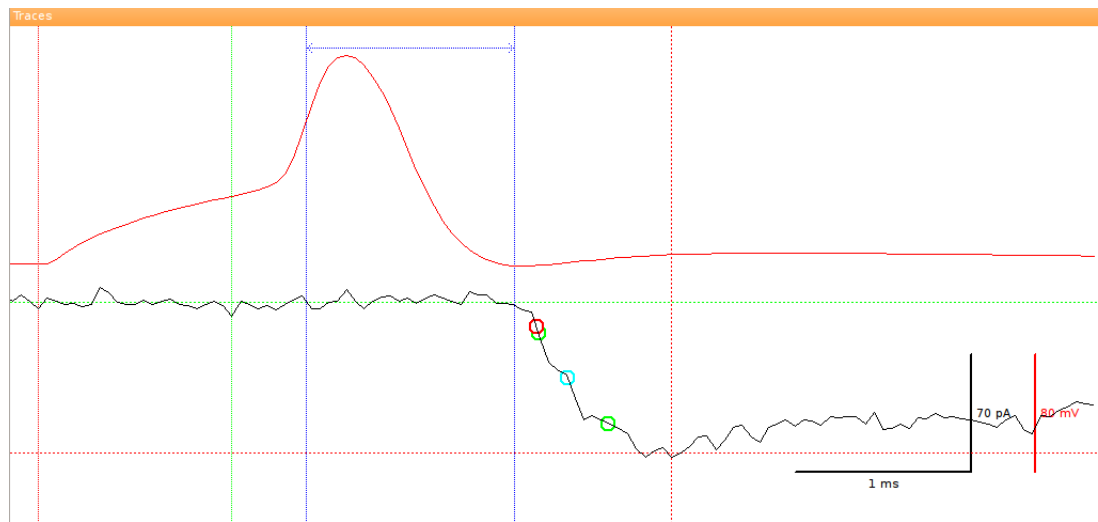
Fig. 20: The latency between the maximal slope of rise of an action potential (red) and the foot of an EPSC (black) is indicated by a horizontal double-headed arrow.