AutoRsComp

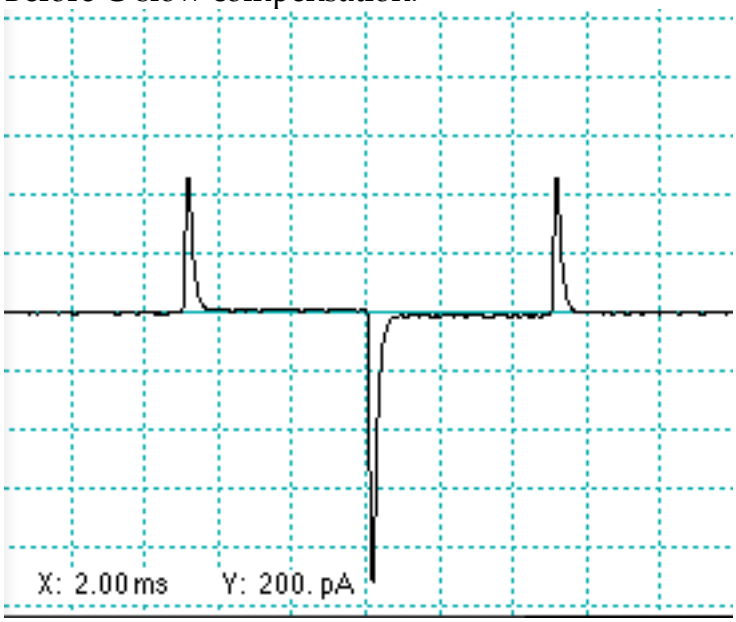
A new procedure to automatically obtain a reasonable Rs-compensation setting has been devised. This new procedure allows to visually verify the correctness of the obtained Rs-compensation value.

How to proceed for the AutoRsComp function

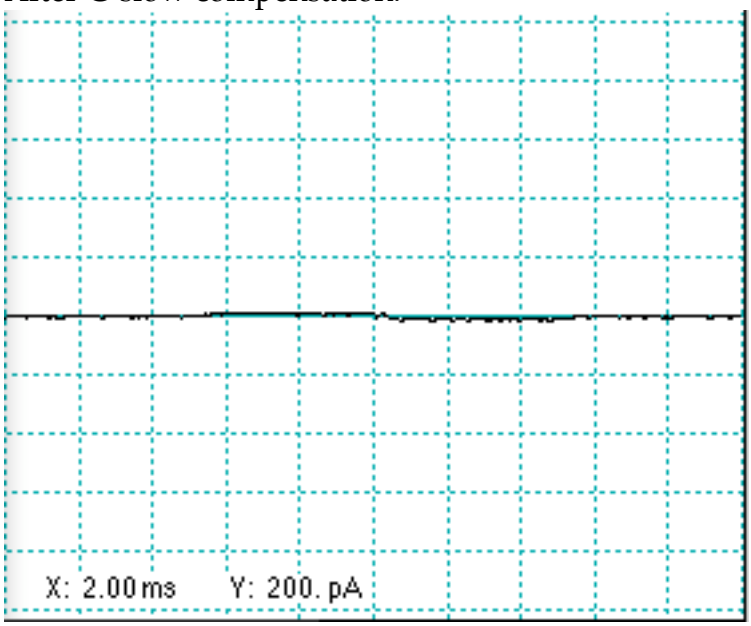
1. Compensate C-fast the usual way.

2. Compensate C-slow the usual way.

   Before C-slow compensation:



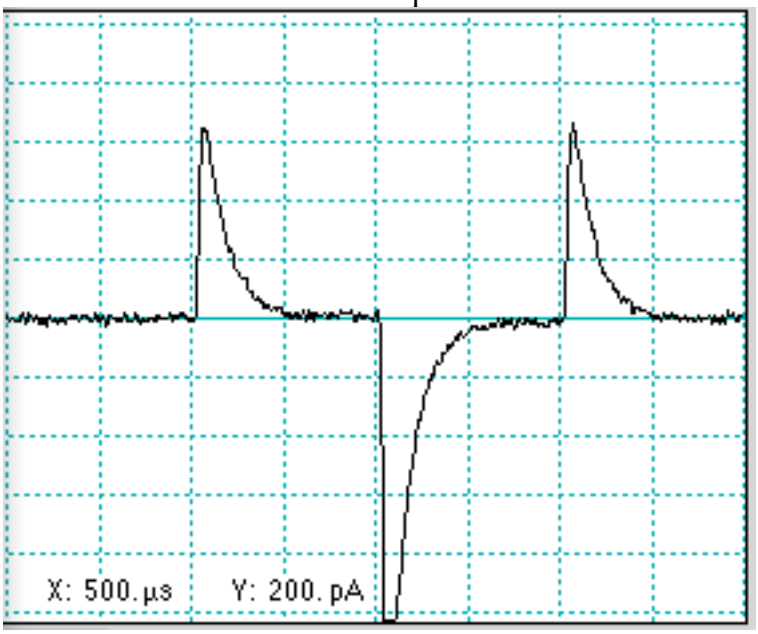
   After C-slow compensation:



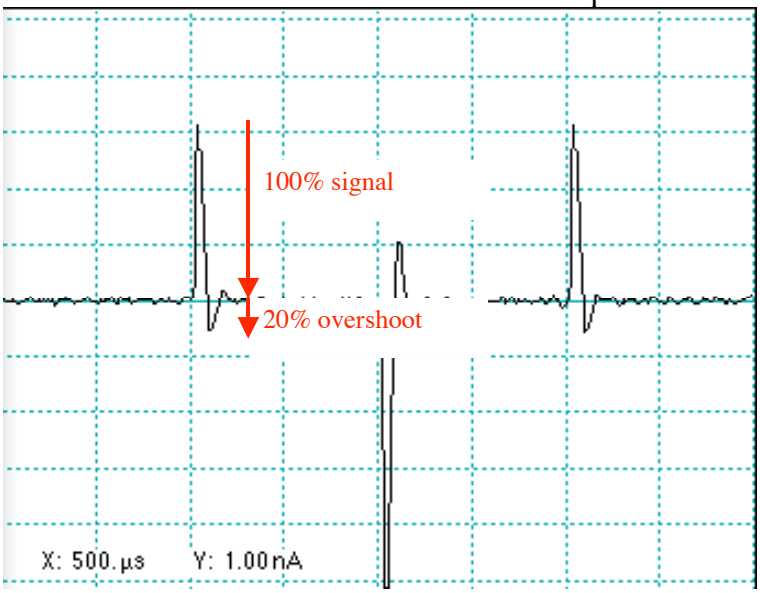
3. Manual equivalent to the Auto-RsComp procedure

   a. Switch C-slow range **off** (range, not value!).
   b. Switch RsMode to 2us.
   c. Switch Input ADC to Imon1.
   d. Set the test pulse amplitude to 5mV and the length to 1.0ms. The resulting sampling interval will be 5us.

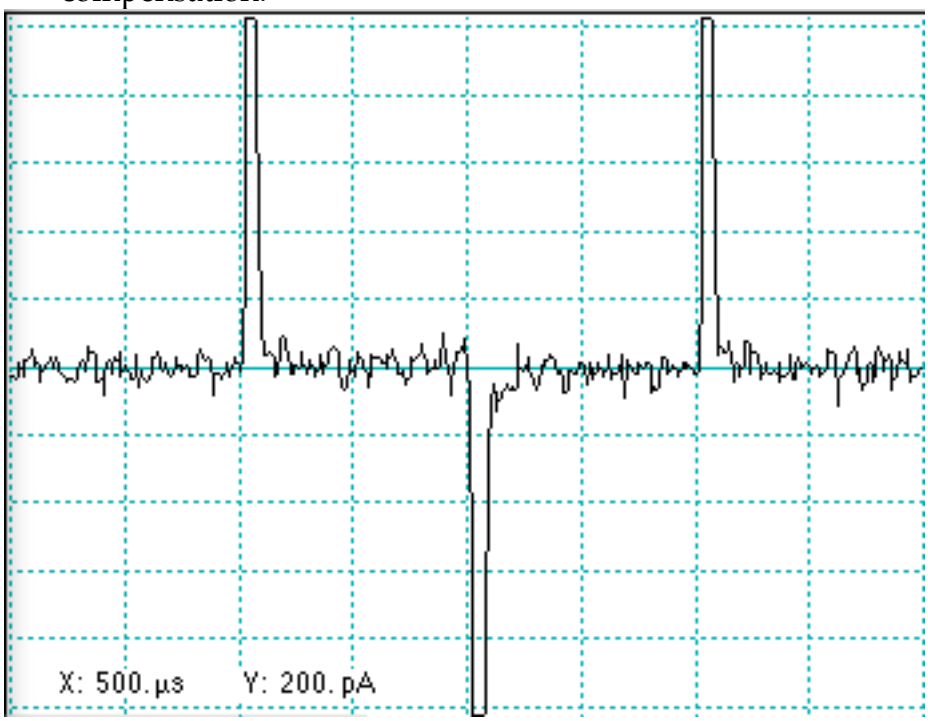
One can now see the un-compensated current trace:



Definitions of used terms on an over-compensated current trace:

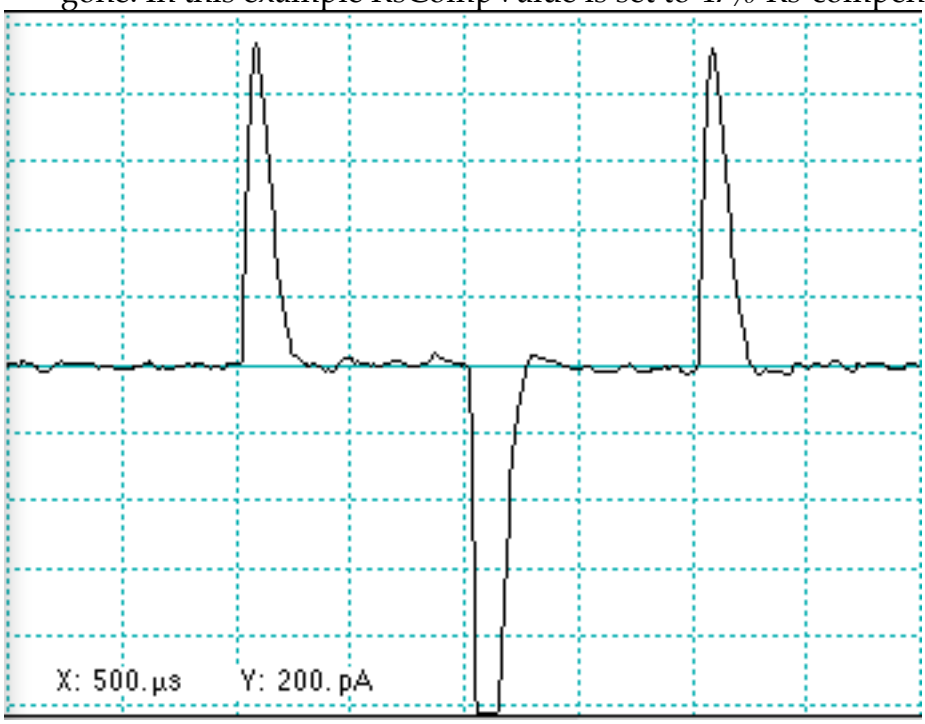


e. Increase RsCompValue as high as possible without that the current response shows an overshoot. In this example RsCompValue is set to 65% Rs-compensation. That is a quite proper setting for correct Rs-compensation:



f. Switch RsMode to 10us.

g. Decrease RsCompValue until the overshoot in the current trace is gone. In this example RsCompValue is set to 47% Rs-compensation.



Actually, the correct approach would be to increase RsCompValue only as long as the response curve remains a decaying exponential. Yet that criterion is much less clear a signal, due to

- the limited number of available points (4 to reach maximum, about 12 for the decay),
- the fact that the response actually is composed of 2 exponential (the filtered rise-time plus the decaying phase), and
- the interference by noise in the response.

2 additional parameters allow to customize the procedure:

a. % Overshoot. This parameter changes the above algorithm to behave similarly to the procedure described by "increase RsComp till oscillation sets in". One could specify e.g. 10% overshoot. The default value would be 0% signal overshoot to obtain the algorithm proposed above.

b. % Backoff. This parameter allows to use a smaller setting that the one obtained by the proposed algorithm. The default value would be 2% reduction of RsCompValue.