Tuning ALTRO's Tail Cancellation for the STAR TPC

Tonko Ljubicic (BNL)

Quick Introduction

The STAR TPC uses the ALICE/CERN developed ALTRO ASIC to provide digital tail cancellation as well as zero suppression in the TPC frontend.

The zero suppression is parametrized with 2 constants: the threshold which the ADC sequence needs to pass and the count of those sequences. Typical values used in the STAR TPC are:
threshold>=3 and sequence count>=1.

I.e. a sequence of ADC values along the timebin:

1 2 3 2 2

will be passed as

0 0 3 0 0

The tail cancellation filter is a digital 3 pole filter with 6 constants K[3] & L[3] which are unsigned short integers in the range from 0-0xFFFF.

For more information please consult the ALTRO manual at http://ep-ed-alice-tpc.web.cern.ch/ep-ed-alice-tpc/doc/ALTRO_CHIP/UserManual_draft_02.pdf.

Tail Cancellation Tuning

As a first step we propose to obtain the best values of these 6 parameters using Yuri Fisyak's TPC simulator which was tuned and matched to the actual TPC data while the tail cancellation was switched OFF.

We obtained the software simulator for the ALICE chip from the ALICE group and we use it to adjust the filter constants by running the simulator over the input data and compare the output obtained with the required performance (shape).

Once we have a set of parameters which satisfy our needs based upon the TPC simulator we plan to evaluate and cross-check with the real TPC events taken in FY09.

What we aim for

We aim to optimize 3 things at once:

1) The total width (in timebins) of the filtered output needs to be as small as possible.

This is the main purpose of the filter -- to cancel the tail and provide better two-track resolution in the time (aka z) direction. However, this value should not be too small (i.e. not less than 3 timebins for a MIP) because we will lose the time resolution.

2) The shape of the pulse leading to and including the highest ADC count should be left unchanged.

The so called "attack" part of the shape should be left intact by the filter. This is the part of the curve which contains the best charge information and which has the best S/N ratio. Admittedly, this is pure heuristics and perhaps a better measure of charge and S/N can be devised.

3) The undershoot (or overshoot) of the filter should be kept to a minimum.

Typically, tail suppression circuits may cause large undershoots if they are mistuned (too much tail cancellation). Conversely, the tail might cause overshoots (too little tail cancellation).

Technique

Input Shape Analysis

For every input shape we determine the start of the pulse (adc_start), the position of the peak (adc_peak) and the end of the pulse (adc_end), all in timebins.

1) We sum up the charge from [adc_start, adc_peak] and we call this the "attack" charge.

2) We count all the timebins where the ADC>=3 and we call this the "width" of the pulse.

3) We sum up all absolute values of the charge (because it might go negative in case of undershoot) from [adc_start+10, adc_end] and call this the "overshoot" of the pulse. We chose to only "look" at the tail after 10 timebins because we believe there is no point in trying to optimize for very closely spaced hits (those less than 10 timebins apart). However, this is again a heuristic which can be changed.

Output Shape Analysis

For every set of the 6 parameters we run the ALTRO simulator on this input data, obtain the output shape and we calculate:

1) the difference between the "attack" of the input and output and require it to be as small as possible.
2) the "width" of the output and require it to be is as small as possible
3) the "overshoot" of the output and require it to be as small as possible.

Minimization Procedure

Crafting a good minimizer which will give us the 3 requirements above is not straightforward. Especially because there is strong interplay between them. As a first shotgun approach a minimizer consisting of stepping through all possible K/L values with a coarse stepsize (i.e. 2000) followed by a simplex minimizer was used to obtain the following values for one specific shape:

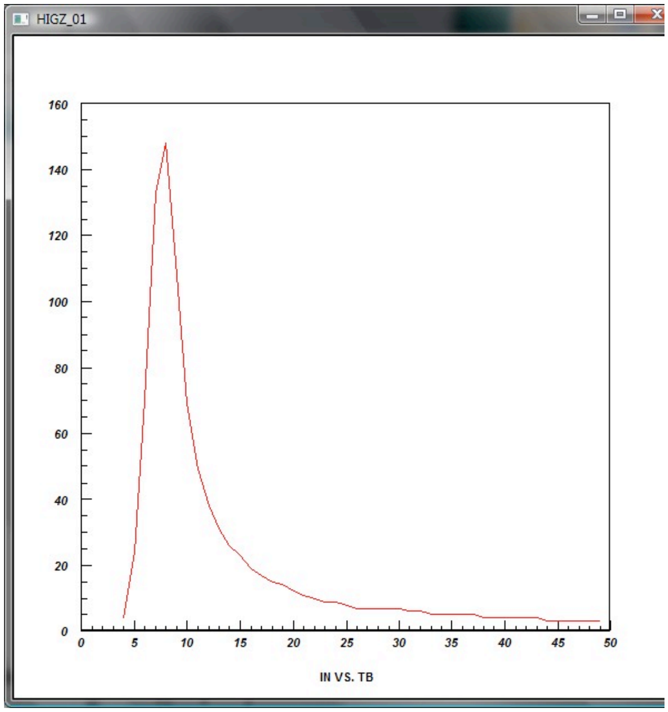

Figure 1: A typical input shape scaled to a highly ionizing particle (about 7 MIPs). Note that the width of this pulse if 46(!) timebins.

Using the heuristical minimization technique above one can obtain a pulse shape with the following characteristic:

width = 9
attack difference = 5
overshoot = 12

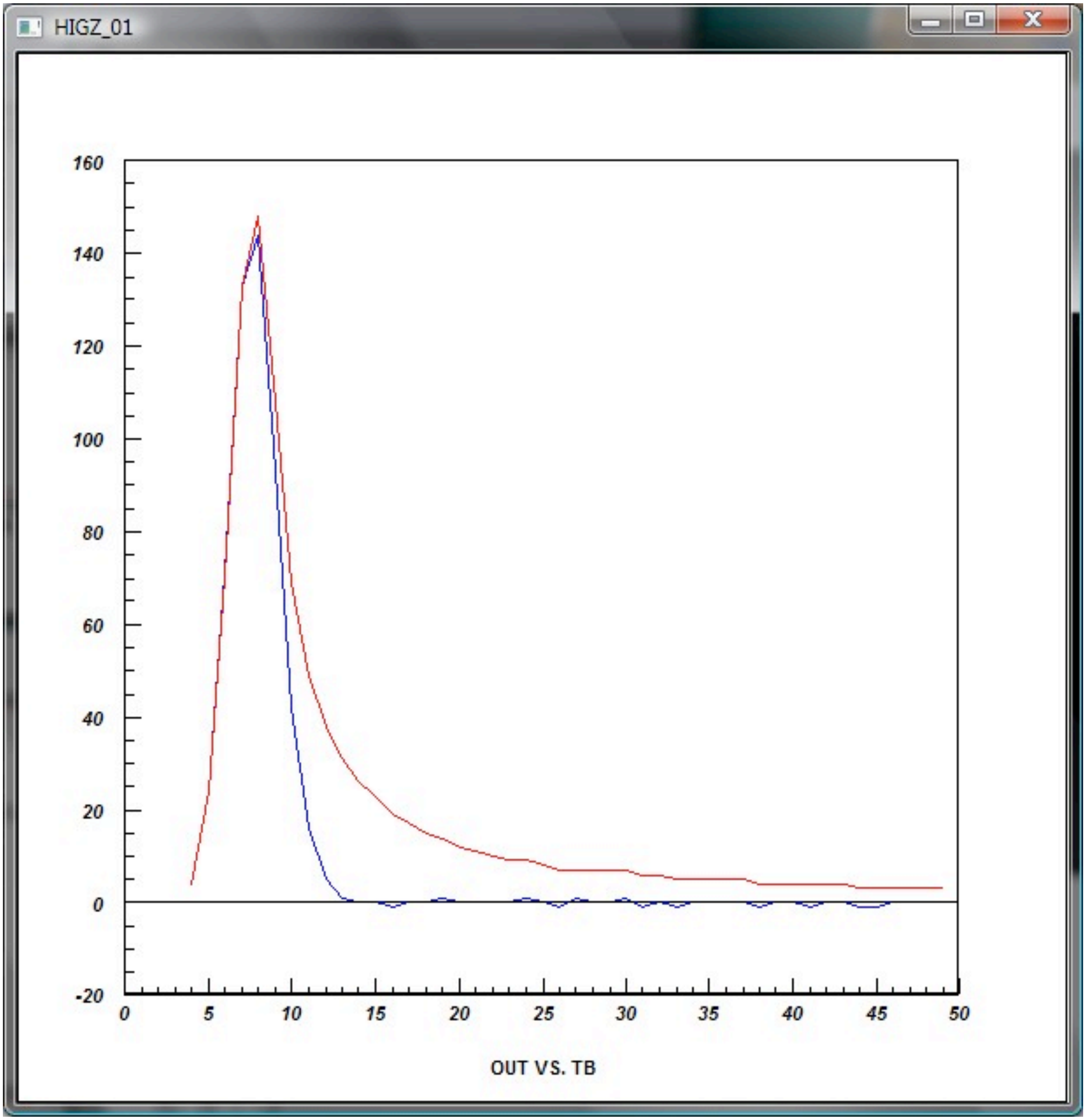

Figure 2: The obtained pulse is shown in blue, the original is shown in red.

The example code can be found in STAR's CVS in online/RTS/src/ALTRO_EVAL (altro_tests.C). It uses the GNU Scientific Library (GSL) for the simplex minimization.