Simulation Waveform Calibrations

L.P. Alonzi

September 5, 2011

1 Assess Damage

You really have to get your hands dirty to understand what's going on. Get yourself some data (both flavours) and then in wffit.c toggle "showTgtWaveforms" to true and replay 1000 events of each kind. Click through the resulting pictures one by one to get a feel both for where the measurement and fitting is at as well as where the simulation is deficient.

name of waveforms in root file: tgt_filt_XXXX

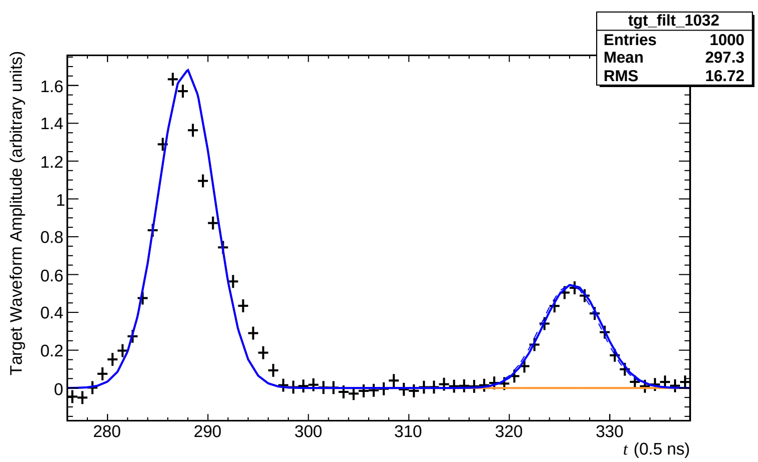

Figure 1: caption

At this point several problems are likely to crop up. First I will discuss a list of possible problems by explaining the waveform production and analysis pipe line. The waveforms are stored completely in RDGT bank. The corresponding routine in the simulation software is "fill_RDGT_bank(...)". The inputs from geant that this function needs are

now there is some question as to whether the geant values can contain an error. of course they can but the design is such tha tthe geant values are so low level they are very unlikely to contain an error. In this case we see that the

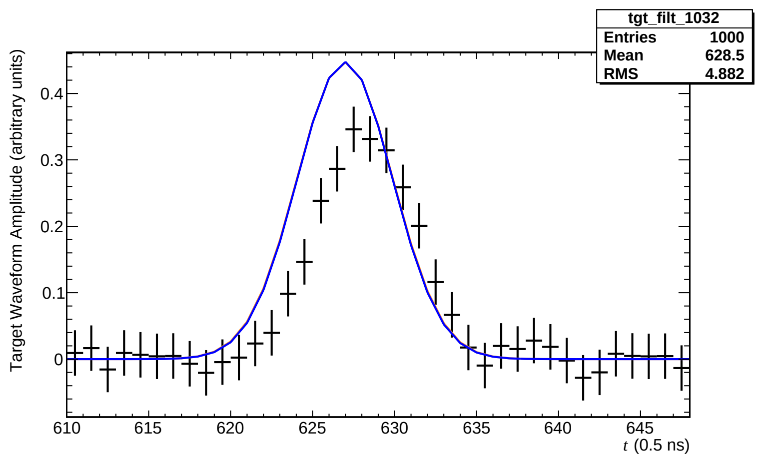

Figure 2: caption

eKin[8]
Edep_Wdeg_L
Edep_Wdeg_R
Edep_Wdeg_T
Edep_Wdeg_B
Edep_Pi_B0
Edep_Pi_DG
Edep_p_Tgt
Edep_Mu_Tgt
Edep_e_TgT
Edep_B0
time[1]
time[6]
time[12]
time[13]
TUBE_mu_t
TUBE_e_t
TUBE_pi_t

Table 1: caption here

geant values come down to energy depositions, times, and one energy check for DIF.

The theory for production of the simulated waveforms is as follows. Every waveform is caused by a charged particle travleing through some active material. That produces scintillation light which is picked up by a photomultiplier tube and then digitized. We assume the characteristic shape of the waveform is identical for all particles and that the energy deposited sclaes linearly with the amplitude (not integral) of the pulse (see leo for that relationship).

To simulate this procedure we need a time and an energy for each pulse from geant. We also need a prototype waveform to build ours from. In the simulation we fill the exact same strcutre as measurement data which means a 1000 bin waveform with 0.5ns per bin. The prototype waveform is binned at teh 500fs level enabling placement of the simulated waveform at the 0.1

Using the measured time the waveform is placed int he correct positiona nd using the energy value is scaled to the appropriate value. Finally a poisson smearing is applied on a bin by bin basis to mimick the photo electron statistics.

The most likely errors observed will be prediction times being incorrect and prediction amplitutes being incorrect. However there are several methods of being wrong. There will be an offset and a "random fluctuation". The next section will address how to attach each of these problems.

2 Identify the problems

It is important that this step be tackled in the correct order because somet thing are dependent on other in the analyzer. The following list shows the symptom followed by the problem followed by location of fix

Pion times fluctuate - incorrect pion time prediction - whatever the prediction reads Pion times offset - incorrect pion time offset - pion time offset parameter in fill_RDGT_bank(...) Pion amplitudes fluctuate - incorrect pion energy predictiosn - whatever the predcition reads

The muon is special because it's time is left free to vary, therefore problems in the muon are limimted to amplitude and should be fixed simply by scaling the muon waveform appropriately. Muon amplitude - should always be "4.1MeV" - adjust scale in fill_RDGT_bank(...) if mismatch

The positron suffers from much worse predcitions that the pion. This is because it is easy for the positron to scatter and there is a vertex walk introduced by the muon decay. Therefore only try to get to the precision of the measurement.

Positron time fluctuate - incorrect prection - whatever the predcions use Positron time offset - positron time offset - adjust parameter in fill_RDGT_bank(...) Positron amplitude fluctuate - again predictions - whatever the predictions use

3 formulate the attack plan

From the analysis of step 2 the necessary parameters to adjust should now be known. Look them up on the table below.

the order of operations is as follows

1. degrader energies
2. b0,degrader time difference

3. pion and positron offsets
4. fine tuing the offsets (special calibration pass4)

4 make the changes

Now it's time to actually adjust the parameters. use the SVN system to record your progress also save all of the data you use as you go. One measurment file is enough as is 1 simulation pi2e and the corresponding ammount of michel files. This step is repeated until each progressive step is within tolerances. It is important to know what your figure of merit is before you begin. Additionally there are some macros already produced to assist the user. The list is incomplete and has some overlap, hopefully that will get cleaned up at some point.

compare_X3X2.C compare_tgt_xyz_trak.C compare_tgt_xyz_prediction.C compare_pion_tof.C compare_pathlength.C compare_digi_times.C mwpc_alignment.C compare_degrader_energies.C calibrate_pion_waveform_1934.C look_at_predictions_1934.C

Each important parameter follows the following (heh) formula

$$\pi_{amp} = \left[f(\mathrm{digi} \to \mathrm{t\_deg} - \mathrm{digi} \to \mathrm{t\_b0}) - \sum \mathrm{digi} \to \mathrm{e\_deg\_i} \right] \tag{1}$$

$$\pi_t = 2 \cdot [\mathrm{digi} \to \mathrm{t\_deg} + \mathrm{f}(\mathrm{getpionenergy}(\mathrm{i}, \mathrm{pevent}) - \mathrm{digi\_etotaldeg}(\mathrm{i}, \mathrm{pevent}))^{-1}] \tag{2}$$

$$\mathrm{e_t} = 2.0 * (\mathrm{pvet} \to \mathrm{pv\_tdc}[] - \theta_{\mathrm{correction}}) + \mathrm{POSITRON\_MEAN\_WFOFFSET}; \tag{3}$$

$$\mathrm{e_{amp}} = \mathrm{f}(\mathrm{trak} \to \mathrm{p\_tgt}[0]) \tag{4}$$

cross reference with table blerg from section blah to amek sense. note that the positron amp requires detailed pathlength configuration and at this time we are not going into that complexity but could if really necessary.

parameter correspondance table

analyzer	simulation
digi->t_deg	pi_dg_index
digi->t_b0	pi_b0_index
digi->e_deg_l[i]	deg_scaleL
digi->e_deg_r[i]	deg_scaleR
digi->e_deg_t[i]	deg_scaleT
digi->e_deg_b[i]	deg_scaleB
pvet->pv_tdc[]	
trak->p_tgt[0]	

Table 2: caption here

Special note about pass4: when using pass 4 you modify the analyzer command to be -N 4 -O yreplays/result84700_1934.codb and make sure to use the pass 3 odob files, the vairables to compare are

they should match the corresponding odb file for measurement data.

A pi gain	=	DOUBLE	:	0.912707
A_pi_gain_ERR	=	DOUBLE	:	0.084006
A mu gain	=	DOUBLE	:	0.962876
A_mu_gain_ERR	=	DOUBLE	:	0.041691
A e gain	=	DOUBLE	:	0.944518
A_e_gain_ERR	=	DOUBLE	:	0.235967
t pi tgt alt offset	=	DOUBLE	:	-0.805893
t_pi_tgt_alt_offset_ERR	=	DOUBLE	:	0.078343
t mu tgt alt offset	=	DOUBLE	:	0.007391
t_mu_tgt_alt_offset_ERR	=	DOUBLE	:	0.142190
t e tgt alt offset	=	DOUBLE	:	0.306180
t_e_tgt_alt_offset_ERR	=	DOUBLE	:	0.918560

Table 3: caption here

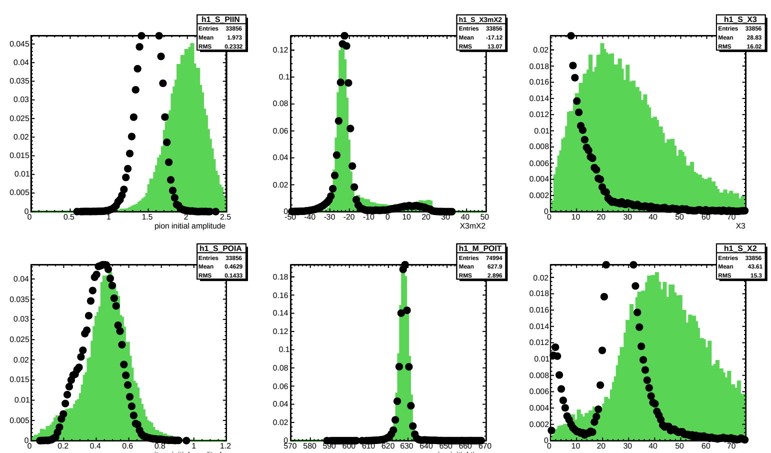

Figure 3: parameters before

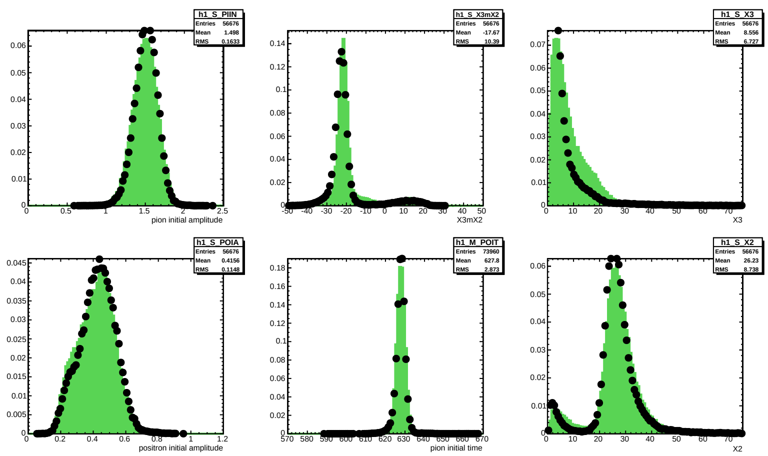

Figure 4: parameters after