hit-time reconstruction study

**Aim**
- Construct methods/algorithms to estimate hit-time from 3 ADC samples
- Check the performance of obtained hit-time by applying them to 6-sample data

### 3-sample time algorithms

**CoG3**

Take a weighted mean of sample timing 
\[
\text{CoG3} = \frac{\sum_{i=0}^{2} a_i t_i}{\sum_{i=0}^{2} a_i}
\]

(We assume that peaking time is independent of amplitude)

**Least Squares (ELS3)**

First, approximate the signal waveform with RC shaper response
\[
a(t) = A \left(1 - \frac{t}{\tau_{SVD}}\right) \exp\left(1 - \frac{t}{\tau_{SVD}}\right)
\]

with peaking time constant \(\tau = 55\) ns

Then, fit with least squares method and determine \(\tau_{SVD}\)

(analytically solvable)

Result: SVD hit-time distribution

(calibrated) hit-time

- Signal peak vs BG tail
  - BG separated
  - different BGD distribution
  - different calibration
  - subtracting \(T_{SVD}\) (analytically solvable)

- Removal of trigger jitter
- Better separation
- Intrinsic resolution*: 2.35/2.28 ns for CoG3/ELS3
- Periodical structure every 12 ns in BGD distribution
- Bunch fill pattern visible

Conclusion

- 3/6-sample mixed mode is now ready and being tested
- 3-sample time algorithms are available in analysis software
- Results are promising, further study will be done with 3-sample data acquired in the test

Use of hit-time (future)

- Use Monte Carlo simulation with various amount of beam background and apply cut on hit-time (CoG3)
  - [−35, +35] (width 70 ns) cut
  - Signal hit efficiency: 99.55(2)%
  - Background hit rejection: 80.89(6)%
  - Higher tracking efficiency, lower tracking fake-rate

**Data readout of SVD**

1 ladder of SVD

- Trigger
- Data processing with FIFI @ 31.8 MHz
- Readout 3 or 6 ADC samples per trigger (currently 6)

**In future**

- Achieve higher luminosity 
- More background hits

**Background hit rejection**

- Physics signal hit: collision bunch beam background hit: all bunches
- Cut on hit-time (= bunch pos.)

**Pick up 3 samples from 6**

We have 6 samples. We want 3 samples with:
- a. largest ADC sample -> amplitude
- b. rising edge* -> hit-time

To pick up such 3 samples offline**, we take successive 2 samples with the largest sum (a) and 1 sample before them (b) (with the best hit-time resolution among the candidates).

*edge is steeper in rising than falling
**In a real implementation, we only tune the latency (waveform unavailable)