



BELLE2-NOTE-PL-2018-005
DRAFT Version 1.0
May 23, 2018

Early Plots on SVD Reconstruction Performances

G. Casarosa*

Università di Pisa, Italy

*Electronic address: giulia.casarosa@desy.de

1. INTRODUCTION

This note contains plots showing SVD reconstruction Performances on early collision data. It is meant to provide information for reproducing these plots, to be approved for showing very early results at conferences.

2. DATASET

The plots in this note have been produced using Experiment 3, Run 783 raw data, located at kekcc at the following location: `/hsm/belle2/bdata/Data/Raw/e0003/r00783/sub00/`.

The run contains 113290 events, all events of the runs have been use, no particular skim has been applied.

2.1. Time Structure of the Bunches

It is important to report the time structure of the bunches for this run, as it has direct implication on the cluster time. the machine allows for 5120 buckets distant roughly 2 ns one another. During the data taking of run783 of experiment 3, 600 buckets where filled, as shown in Figure 1. The bunch interval is therefore ~ 16 ns.

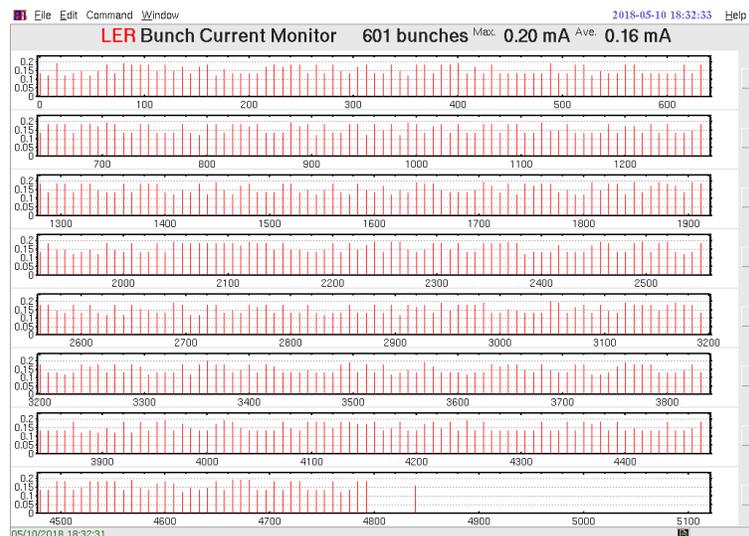


FIG. 1: Machine Filling Pattern when the Exp3 Run783 was taken.

Since the APV25 readout clock is synchronized to the radiofrequency of the machine, and it has a period of ~ 32 ns, we know that we have two bunch crossings in each APV clock, as shown in a simplified cartoon in Figure 2.

3. RECONSTRUCTION AND SELECTION

The events have been reconstructed with the standard reconstruction of the master release (git commit `77aefe63e8b72f8a8b342b23b91dfb9f8a4c9b02`). The software version used does

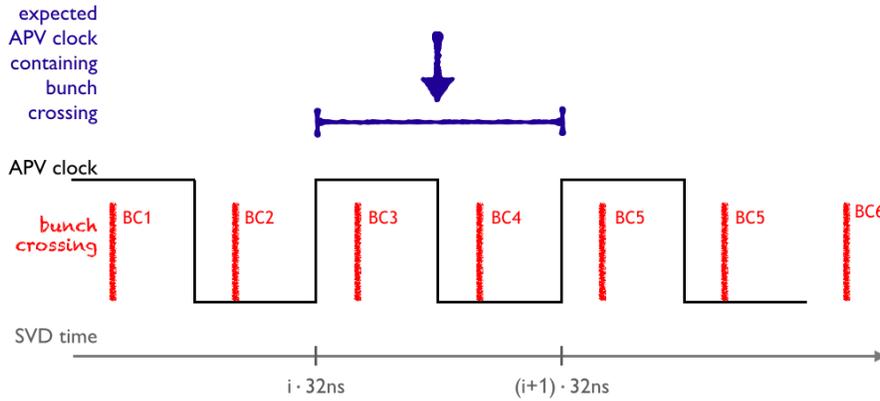


FIG. 2: APV clock and Bunch Crossing time structure.

not differ from the one or release-01-02-03 that is currently used for reprocessing, in particular concerning the svd and tracking modules and scripts affecting the standard reconstruction. The data is reconstructed using the Calibration_Offline_Development open global tag.

The standard reconstruction steps are the following:

- SVD Reconstruction

1. for luminosity runs the zero suppression cut applied online is 5. This means that only strips with at least one sample with an amplitude greater than 5 times the noise are usable for offline reconstruction. The other strips are suppressed on the FADC.
2. Strip Charge estimated as the largest among the 6 samples. Strip Time estimated as the average of the times of the 6 samples, weighted with the amplitude of each sample. This average is corrected by the peak time of each strip estimated from local calibration runs.
3. Strips are combined into clusters: nearby strips are assembled in a cluster. The online zero suppression cut at 5 assures that all strips received or offline reconstruction have enough signal to be attached to clusters. The cluster charge is the sum of the charges of all strips, the cluster time is the average of strip times weighted with the strip charge.
4. SVD space points are built as all combinations of U and V clusters belonging to the same sensors.

- CDC Reconstruction (no details provided)

- Pattern Recognition: SVD hit time is not used

1. CDC pattern recognition (no details provided)
2. CKF pattern recognition: extrapolation of CDC tracks towards SVD sensors and attachment of additional hits to the track candidate
3. VXDTF2 pattern recognition: SVD-only pattern recognition, run on the remaining SVD SpacePoints

- Track Fitting: track candidates are fitted to extract the track parameters.

After the reconstruction, we select only clusters related to tracks. We do not apply any selection on the tracks.

4. PLOTS

In Figure 3 we report the Signal-Noise ratio for clusters related to tracks, for the large rectangular sensors (all backward and origami modules), distinguishing between U/P and V/N side clusters. The noise of each cluster is computed as the sum in quadrature of the noise of the single strips, while the cluster charge is the sum of the charges of the strip forming the cluster. Since the cluster noise depends on the cluster size, we report the most probable value of the SNR distribution distinguishing different cluster sizes in Table I. The plots of the distributions of SNR for different cluster size will be reported in Section 5. The majority of the clusters are composed of two strips, as expected.

cluster size	fraction	U-side SNR	V-side SNR
1	15%	22	32
2	60%	22	24
3	20%	30	28

TABLE I: Signal-to-Noise Ratio most probable values for U and V clusters related to Tracks, for the large rectangular sensors. The fraction of clusters for each cluster size is also reported.

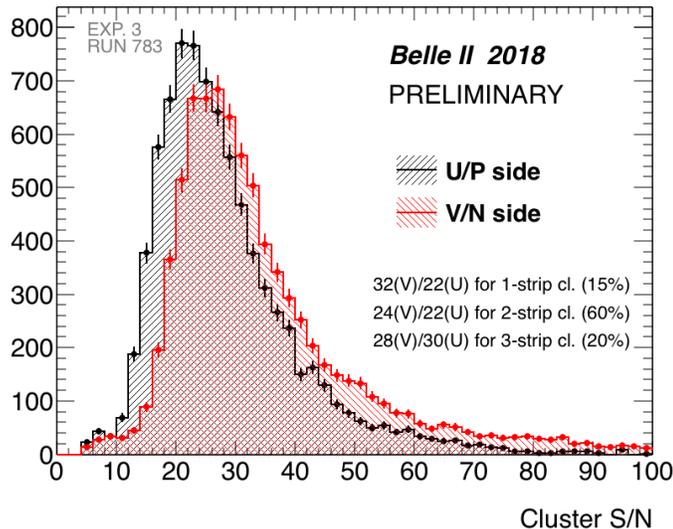


FIG. 3: Signal-to-Noise Ratio for clusters related to Tracks, belonging to all Large Rectangular Sensors.

In Figure 4 we report the reconstructed time of V/N clusters related to tracks, for the large rectangular sensors (all backward and origami modules). The peaking structure can be seen

completely understood looking at Figure 2. The two main peaks in the middle correspond to clusters created by tracks belonging to the bunch crossings that has happened in the APV clock period expected from the Trigger signal (BC3 and BC4). In particular, the bunch crossing BC3 corresponds to the peak at -45ns and the BC4 corresponds to the peak at -35ns. The distance between the peaks is a bit smaller than the expected 16ns, this is actually expected and verified on simulation, since the CoG Time estimator is not a linear estimator. The two smaller peaks correspond to events in which the trigger had a jitter large enough to miss the right APV clock by one. In particular, the peak on the left at -65ns correspond to events actually happened in the bunch crossing BC2, while the peak at -25ns corresponds to BC4. The absolute value of the reconstructed time is not yet calibrated, therefore only relative times are meaningful at this stage.

This picture show the very nice performance of SVD in determining the cluster time. First of all we are able to distinguish clusters belonging to tracks from events that are 16ns apart, also if the trigger arrives with a significant jitter. This means that we may participate in determining the event time with respect to the trigger signal. The width of each peak give us indications on the SVD hit timing resolution that, although at a very early stage of calibration, is well below 10ns, as expected.

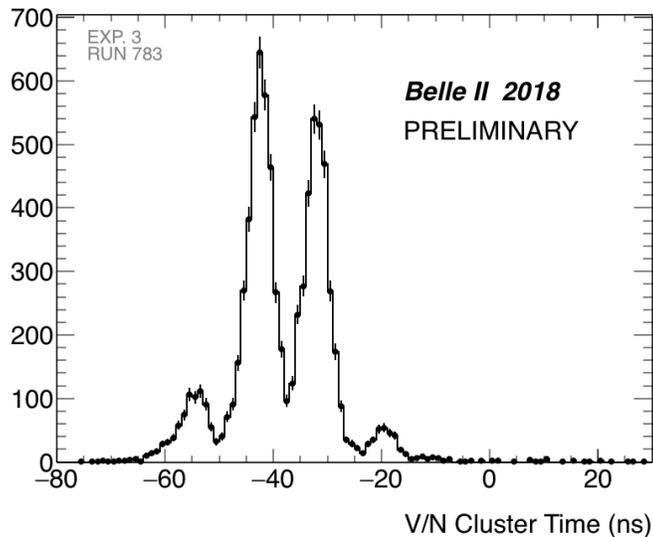


FIG. 4: Reconstructed time for V-clusters related to Tracks, belonging to all Large Rectangular Sensors.

5. ADDITIONAL PLOTS

6. SCRIPT

Here we report the script used for the reconstruction, the module that produce the side-dependent plots (SVDPerformance) is also appended to the path.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import os
from basf2 import *
from ROOT import Belle2
from tracking import add_tracking_reconstruction
from rawdata import add_unpackers
import sys

print('***')
print('*** this is the reconstruction script used:')
with open(sys.argv[0], 'r') as fin:
    print(fin.read(), end="")
print('*** end of reconstruction script')
print('***')

run = int(sys.argv[1])
tag = 'data_exp3_Run'+str(run)+ '_pub'
folder = 'Run'+str(run)+'/'

shapers = "SVDShaperDigits"

# setup database
reset_database()
use_database_chain()
use_central_database("Calibration_Offline_Development", LogLevel.WARNING)
#use_central_database("GT_gen_data_004.51_reprocessing-release-01-02-03", LogLevel.WARNING)
#use_central_database("MagneticFieldPhase2-2018-04", LogLevel.WARNING)

# Create paths
main = create_path()

main.add_module('RootInput', inputFileNames=[str(folder) + 'rootOutput/*root'])

main.add_module("Gearbox")
main.add_module('Geometry', useDB = True)

# SVD+CDC unpacking
add_unpackers(main, components=['CDC'])
main.add_module('SVDUnpacker')

# SVD reconstruction
```

```

recos = 'SVDRecoDigits'
clusters = 'SVDClusters'
fitterName = 'SVDCoGTimeEstimator'
clusterizerName = 'SVDSimpleClusterizer'
goodfitter = register_module('SVDCoGTimeEstimator')
goodfitter.set_name(fitterName)
goodfitter.param('Correction_TBTimeWindow', False)
goodfitter.param('Correction_ShiftMeanToZero', False)
goodfitter.param('ShaperDigits', shapers)
goodfitter.param('RecoDigits', recos)
main.add_module(goodfitter)
goodclusterizer = register_module('SVDSimpleClusterizer')
goodclusterizer.set_name(clusterizerName)
goodclusterizer.param('RecoDigits', recos)
goodclusterizer.param('Clusters', clusters)
main.add_module(goodclusterizer)

#add tracking reconstruction
add_tracking_reconstruction(main, components=['SVD', 'CDC'])

#remove cut on CoG in SpacePointCreator
for moda in main.modules():
    if moda.name() == 'SVDSpacePointCreator':
        moda.param("MinClusterTime", -999)

perf = register_module('SVDPerformance')
perf.param('ShaperDigitsName', shapers)
perf.param('RecoDigitsName', recos)
perf.param('ClustersName', clusters)
perf.param('outputFileName', str(folder) + 'SVDTRKPerformance_' + str(tag) + '.root')
main.add_module(perf)

print_path(main)
main.add_module('ProgressBar')

# Process events
process(main)

print(statistics)

```