Table of Contents

VELO Online Monitoring Wish List 1
Data retrieved from PVSS 1
Detailed Specifcations 1
Considered 1
Available measure quantities from hardware 1
Correlations of some of these 1
Radiation monitoring 1
Power supply values 1
BEETLE values 1
TELL1 values 1
Data derived from analysis jobs 2
Low frequency monitoring 2
Detailed Specifications 2
Higher level quantities 2
Alignment/Resolution (Marco, Silvia) 2
TELL1 Algorithm Monitoring, require NZS data (Tomasz, Chris) 3
Considered 3
Quantities derived from NZS Data 3
Quantities derived from ZS data 3
Higher level quantities 3
High frequency visualization 4
Detailed Specifications 4
Considered 4

VELO Online Monitoring Wish List

This page is meant to be a scratch area for collecting ideas what to monitor and how. So everybody should feel free to add some ideas! As a starting point I have added a few items collected from various sources.

-- Kurt Rinnert - 13 Mar 2008

Data retrieved from PVSS

In this section we collect quantities we want to monitor that can be directly retrieved from PVSS.

Detailed Specifcations

None yet.

Considered

Available measure quantities from hardware

- Si current
- Beetle currents
- Temperature of hybrid
- Rack temperatures
- Temperature of RF foil
- Beam currents

Correlations of some of these

- Si current vs temp.
- Beetle current and Si current superimposed
- Current vs foil temp.

Radiation monitoring

- Measure Si current
- BCM
- 2 beam currents

Power supply values

- Current and past history

BEETLE values

- Number of resets per Tell 1.
- Pipeline synchronisation errors.

TELL1 values

- Need to know how often the NZS bank is read out (approx once/s).
- Need to decide on list of variables to monitor.
- Want to track pedestal values.
- RMS of common mode.

Data derived from analysis jobs

In this section we collect quantities we want to monitor that are derived from analysis jobs. More specifically everything that is based on results of a Gaudi job running on the the online farm. There are two sub-categories: slow monitoring with history and fast monitoring without history. We expect the slow monitoring to be updated with ~0.01 Hz and the fast monitoring with ~2 Hz. For the fast monitoring/visualization it is forseen to run a dedicated job on a dedicated node, circumventing the client/server architecture that is used for the standard, low frequency, monitoring.

Low frequency monitoring

Detailed Specifications

Higher level quantities

Alignment/Resolution (Marco, Silvia)

- residuals (preferably for small angle tracks with many spacepoints)
    - vs r/phi for alignment
        - for each sensor make unbiased straight line track fit and compute intercept with sensor
        - calculate residual (`DeVeloSensor`)
        - for R sensor: plot residual vs phi (calculating phi from track state)
        - for Phi sensor: plot residual vs r (calculating r from track state) and vs phi (phi of strip at minimum strip radius +/- stereo angle)
        - when histogram has about 10'000 entries at least 1 minute old) create a `ProfileX()` (root method)
        - this **profile** should be kept, update can be ensured to be **>=1min**
        - fit profile vs phi with `par_0 * sin(phi) + par_1 * cos(phi) + par_2`
            - plot distribution of `par_0` and `par_1` for all sensors in one histogram
            - this will be an **overview histogram of x/y misalignments**, update frequency same as profile: **>=1min**
        - fit profile vs r with `par_0 * r + par_1`
            - plot distribution of `par_0` for all sensors in one histogram
            - this will be an **overview histogram of z-rotation misalignments**, update frequency same as profile: **>=1min**
        - (all code available from software alignment)
        - res_Phi_vs_phi.eps: Phi sensor residuals vs phi
        - res_Phi_vs_phi_fitted.eps: Profile and fit of Phi sensor residual vs phi
    - vs beetle readout direction for R sensor x-talk
        - largest cross-talk depends on readout direction and is particularly visible for R sensors
        - Phi sensors are less strongly affected by cross-talk due to different readout pattern
        - for each R sensor plot: residual distribution for each readout direction in each sector, i.e. make 8 plots: 4 sectors, strips 0-127 and 128-511 per sector
        - any non-zero mean is a sign for cross-talk
        - **cross-talk overview plot**: plot the distribution of the 8 mean values per sensor for all sensors, update frequency: **>=1min**
    - vs pitch and angle for resolution
        - for each sensor, plot residual vs pitch and angle (projected angle for Phi sensors?) 3-dim histogram
        - for each bin in pitch+angle fit the spread of the residual distribution
        - the fit should only be done for bins with at least 10000 entries
        - it has to be noted that extrapolation errors contribute to the measured spread!

◊ projections of resolution (spread) vs pitch/angle can be plotted for various bins in the other variable using colour-coded lines (see Tomasz's LHCb note plots)
◊ **residual plots** should not be updated too often to ensure necessary statistics, hence update frequency: **(>)>=10min**

TELL1 Algorithm Monitoring, require NZS data (Tomasz, Chris)

Aim: monitor performance of TELL1 algorithms / check current algorithm parameter values suitable
A separate set of plots are used to tune the parameter values (from the Vetra computers) which are not described here.

- Algorithm:Pedestal Subtraction - Histo 1: mean ADC counts after pedestal subtraction versus chip channel.
- Algorithm:FIR Filter - Histo 1: parameters of cross-talk computer run on data that has already been corrected.
- Algorithm:Beetle Header Correction - Histo1: parameters of beetle correction run on data that has already been corrected, Histo2: corrected rms noise of 1st and 2nd channel of each analogue link / CM corrected rms noise of average channel in same link.
- Algorithm:MCMS Correction - Histo1: noise (rms ADC values) versus chip channel after correction. Histo 2: to be defined something to specifically check events with large baseline swings (JC, Gwen looking)
- Algorithm:Common Mode Correction - Histo1:noise (rms ADC values) versus strip number after correction. Histo2: profile plot of value of correction applied versus strip number.
- Algorithm: Cluster Maker- Histo1: Number of clusters versus central strip number (and number with spillover bit set superimposed). Histo2: Size of clusters versus central strip number. Histo 3: Landau (sum of ADC). Histo4: Signal/Noise.

Considered

Quantities derived from NZS Data

- noise before/after common mode
- pedestals
- ADC vs strip/chip channel
- correlation plots between adjacent sensors
- S/N per sensor (split into regions)
- time development of pedestals
- time development of noise

Quantities derived from ZS data

- hitmaps
- sensor efficiency
- sensor occupancy
- number of clusters vs strip number

Higher level quantities

- tracks (this is very generic, needs to be fleshed out, like e.g. below)
- Alignment monitoring through alignment constants
    - ♦ residuals insensitive to less well constrained DOFs
    - ♦ rerun alignment algorithms and monitor change in parameters

High frequency visualization

Detailed Specifications

None yet.

Considered

- beam position

This topic: LHCb > VeloMoniWishList
Topic revision: r7 - 2008-07-21 - ChrisParkes