COMPUTING BUNCH CHARGE, POSITION, AND BPM RESOLUTION IN TURN-BY-TURN EMMA BPMS

A. Kalinin, G. Cox, J. K. Jones, D. J. Kelliher, S. Machida,
STFC Daresbury & Rutherford Appleton Laboratories, UK
I. W. Kirkman, Liverpool University and Cockcroft Institute, UK
R. G. Borrell, WareWorks Ltd, Manchester, UK

Abstract

The NS-FFAG electron model 'EMMA' and its Injection and Extraction Lines are equipped with a total of fifty EPICS VME BPMs. For each injection, the ring BPMs deliver a snapshot of a turn-by-turn trajectory measured in each of 42 cells. Additional BPMs (two pairs) are used to obtain a turn-by-turn Poincare map. We describe some BPM features, the EPICS architecture, and a set of C#/Python/Mathematica data processing programs. We use pickup mappings that allow get a linear offset response and eliminate bunch charge signal dependence on offset. We present some beam measurement results collected in 2011-12 runs.

INTRODUCTION

In 2011, the EMMA experiment [1] attained a main goal: fast serpentine acceleration with crossing integer tunes, which proved feasibility of NS-FFAG and called for a broader investigation of other topics specific for NS-FFAG in general and for the model 'EMMA' in particular. Most of the tasks need massive trajectory/orbit measurements, which require fast and trustworthy BPMs.

Each two-plane BPM [2] comprises a button pickup, a Front-End (placed in the machine girder) that transmits the signals to a remote VME card which turn-by-turn measures the signals and stores them in a memory. After the last turn, the memory is read by EPICS.

The Front-End has three functions: (1) it converts each button signal to a compact three-wave 700MHz packet; (2) on each turn, it multiplexes two packets of each opposite button pair into one channel as a doublet with spacing $T/4 = 13.8$ns where $T = 55.2$ns is the turn period; and (3) it pre-amplifies the doublet. A Front-End output picture can be found in [2].

On initial stage of the EMMA experiment, the cell-by-cell trajectory/orbit was measured by digitising the turn-by-turn Front-End doublets with a 20GS/s oscilloscope. The signals were averaged over shots. Signal processing was done in computer.

With VME cards and EPICS drivers and processing software developed, faster measurements became possible even with initial word-by-word method of card memory transfer. It was found difficult to make Python processing programs able to acquire and collate the asynchronous data published by EPICS. As a result, some beam offsets returned by the program after a shot, could belong to different shots.

Eventually, a block data transfer mode was implemented in the card, and a new EPICS architecture was developed that could collate and store all BPM readings of a shot in a single array. With the latter done, it became possible to measure trajectory snapshots, which in turn allowed both obtain a true trajectory, and characterise the jitter.

Several BPM cards that are used in the EMMA Injection Line can be swapped to pickups of an ALICE FEL path that is alternative to the EMMA injection path. The BPMs do bunch-by-bunch measurements of long trains (up to 1600 bunches, the bunch spacing is 61.5ns).

BPM VME CARD, BEAM TRIGGER

The two-channel VME card was designed as a measurement station equipped with two sets of a 100MHz 14bit ADC and a 16k memory. As a BPM card, it uses the following mezzanine boards: two Analog Processors (APs), and an ADC Clock (CLK). Main function of the AP is to obtain Front-End packet envelopes that are sampled by the ADC. Besides, one of the APs manufactures a doublet of CLK trigger pulses synchronous with the input doublet.

For each turn, the CLK generates four spaced-by-$T/4$ pulses that clock the ADC. When a no-next-turn event occurs, the CLK stops the clocking, generates an interrupt to the VME interface and sets for reading a number of turns. If a no-beam or single-trigger-pulse-on-the-first-turn event occurs, the interrupt is also generated but no clock is there and the number of turns is set to zero. More details of the CLK work can be found in [2].

In a time-domain multiplexing BPM, a signal crosstalk in the doublet may take place. To take it into account, just four samples per turn are necessary.

Pursued in this BPM, a beam-triggered clock idea is attractive from many points of view, but its implementation brought some problems. Principally, to generate a beam trigger from the output beam signal, a comparator can be used with a threshold that is set as low as, say, 3 times the output thermal noise rms. In a time-domain-multiplexing BPM, this decides range of the measurable beam offset. For EMMA BPM the range as a ratio of the ADC range 2.2V to $3rms$ is 90. However, the $3rms$ threshold limit has not been reached yet. The reason is a double reflection of the button signal first from the Front-End input and then from the button back to the input. For one pickup signal large that takes place with a large beam offset, its reflection signal (that is about 3%

and comes in a time between the doublets) might exceed the threshold and cause a false triggering. To avoid this, the thresholds in all ring BPMs were changed to $12rms$. This is a significant reduction of the range. We expect a radical improvement with a modified AP where the comparator threshold that is normally low, is switched to high (by CLK) for a time interval between the doublets.

BPM CALCULATIONS

BPM Signal Range Setting

The packet envelope at the AP output is a 5ns bell-shape pulse that is sampled at its apex. For each sampling within a turn, each BPM channel gives the voltages:

$$U1 = P + V1 \qquad U2 = P + V2 + \Delta V1 \qquad (1)$$
$$U3 = P + \Delta V2 \qquad U4 = P$$

where P is a DC pedestal, $V1, V2$ are the opposite button signals, $\Delta V1, \Delta V2$ are the envelope tails. The DC pedestal is introduced to allow envelope pulse magnitude cover full ADC range.

For a nominal 40pC bunch the BPM outputs $U1, U2$ for a bunch near the centre are close to a third of the ADC range. For a beam offset close to a half of aperture one of the outputs has the range exceeded. To set the signal back into the range, a digitally controlled attenuator is used built in the AP. With it, it is possible to set the range automatically by a BPM program using several shots. The setting has got successful, if the other, lesser output at the final attenuation is higher than the beam trigger threshold. For the threshold $12rms$ a bunch offset range is about 2/3 of (circular) pickup aperture. It still agrees with an initial EMMA BPM specification but can't provide measurements with an existing injection scheme/tuning where extreme offsets typically take place on first turns.

Button Signal Calculation

We find the button signals as the solutions of the equation system (1):

$$V1 = U1 - U4 \qquad (2)$$
$$V2 = (U2 - U4) - K \cdot (U1 - U4)$$

where the coefficient $K = \Delta V1/V1 = \Delta V2/V2 << 1$ is calculated as

$$K = \frac{(U2 - U4) + \sqrt{(U2 - U4)^2 - 4(U1 - U4)(U3 - U4)}}{2(U1 - U4)}$$

Bunch Offset Calculation

We have two sorts of button pickups, circular pickup with buttons on the x, y axes, and rectangle pickup with two pairs of horizontal buttons symmetrically spaced from the x, y planes. Define each from two pickup own axes $\chi(\eta)$ as a curve where $\eta(\chi) = 0$, with the pickup centre at $\chi = 0,\ \eta = 0$ and an angle between the axes θ. A normalised bunch offset is calculated as

$$\chi_n = \frac{V11_n - V12_n}{V11_n + V12_n}\ ,\ \eta_n = \frac{V21_n - V22_n}{V21_n + V22_n} \qquad (4)$$

where the first index is the BPM channel number, the second index is the button number. Note (4) does not depend on channel gains. For a circular pickup, the axis $\chi(\eta)$ coincides with the $x(y)$ axis, and the offset is:

$$x_n[\text{mm}] = M_c \cdot \chi_n - X_n,\ y_n[\text{mm}] = M_c \cdot \eta_n - Y_n \qquad (5)$$

where M_c[mm] is the pickup scale coefficient, X_n, Y_n is the pickup 'mechanical' zero offset taken from the BPM database. For a rectangle pickup the offset is:

$$x_m = M_r \cdot \sin\theta/2 \cdot (\chi_m - \eta_m) - X_m \qquad (6)$$
$$y_m = M_r \cdot \cos\theta/2 \cdot (\chi_m + \eta_m) - Y_m$$

We use (6) but not the conventional formulae $(V11 - V12 \mp V21 \pm V22)/\Sigma$, as the formulae are not accurate in the case channel gains differ.

Bunch Charge Calculation

In the pickup centre vicinity, the bunch charge q expressed in button voltage units is

$$q = (v11 + v12) + (v21 + v22) \qquad (7)$$

For different channel gains $G1 = G\,(1-\delta), G2 = G\,(1 + \delta), |\delta| << 1$, in the centre vicinity, the charge expressed in BPM output voltage units is equal to

$$q = (V11 + V12 + V21 + V22) \qquad (8)$$

which is similar to (7) with a negligible error $|\varepsilon| << |\delta|$.

For a lossless turn, one can write for N BPMs: $\overline{q} = \frac{1}{N} \cdot \sum_1^N q_n$. Introducing an individual BPM gain as $G_n = \overline{q}/q_n$, a gain-normalised charge at any BPM is:

$$q = G_n \cdot (V11 + V12 + V21 + V22) \qquad (9)$$

Pickup Mapping

For EMMA ring BPMs, we devised and applied a bunch offset mapping that linearises the pickups, and a bunch charge mapping as well, that keeps a button signal sum constant in the aperture. Details of this design are reported elsewhere [4].

BPM Resolution

We planned to apply a beam-based 'quadrupole' combination BPM resolution measurement method. [3] For use of it on the EMMA ring with its large offsets, it was necessary to develop an extremely fine 'quadrupole' combination mapping. Instead, we approached this method on ALICE BPMs where the bunch train can be easily kept near centre.

The BPM resolution can be immediately estimated from Fig. 1 where offset vs bunch charge plots are shown measured in IL BPM2. A large horizontal jitter is due to bunch energy jitter. On a vertical plot, the bars (x10) represent some low bunch jitter on BPM noise background. Each one can be calculated from any pair of the measurements taken beyond the noise plateau (i.e., <40pC). For 20pC the BPM resolution comes to 35µm (expected resolution given in [2] was 30µm). This result agrees with preliminary results obtained on ALICE BPMs using the method above.

ISBN 978-3-95450-115-1

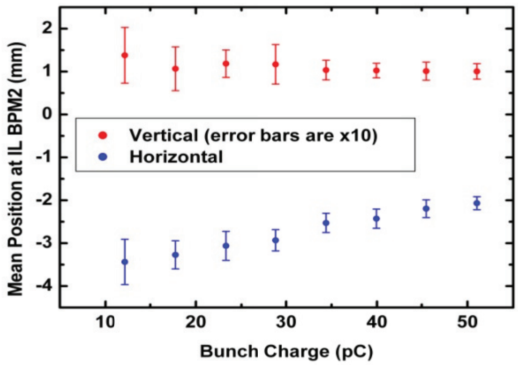


Fig. 1: Bunch offset vs charge.

EPICS ARCHITECTURE

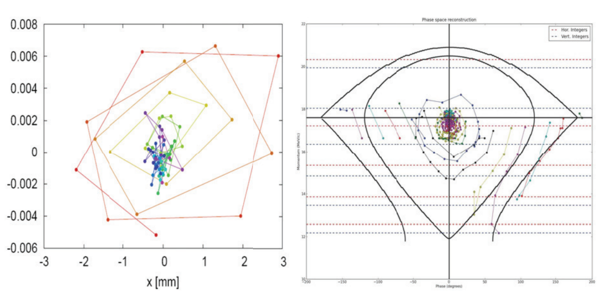
The EMMA BPM system is interfaced to the EPICS Control System via seven VME subsystems (crates) and three 1U rack-mount PCs. The VME subsystems are located on a private GigE network and consist each of a MVME5500 processor running the vxWorks real-time operating system, and up to twenty VME BPM cards. Data is acquired by the EPICS system on I/O interrupts that are generated by BPM cards.

Voltages captured from cards are stored in the VME processor in two's complement format. Only the latest captured voltages are stored in the VME databases, and are overwritten as new data arrives. One Linux PC provides proxy access to the VME subsystems for setting and reading configuration data. Another Linux PC converts the raw binary data to real voltages and makes the data available via EPICS on the EMMA control network. The third PC runs MS Windows and collects and collates the voltages from each shot in single array and archives all the data to a disk for offline analysis.

EPICS BPM performance is 4k turns @ 5Hz across all of 50 BPM cards.

DATA PROCESSING PROGRAMS

A Python/EPICS program module is implemented in the control room to access the BPM data and display orbit. Several programs were written. One of them does real time Poincare map plotting (Fig. 2a) which shows the phase space position of the injected beam and is helpful for injection tuning. Another one displays orbit position as a function of consecutive cells. The orbits are stored with time stamps and are analysed later for COD and tune calculation. In Fig. 2b, longitudinal phase space reconstruction is shown, where the bunch energy was measured turn-by-turn with horizontal BPM.



a) b)

Fig. 2: a) A Poincare map. (Courtesy of J. S. Berg) b) EMMA longitudinal phase space reconstruction (1MeV/turn). On each turn, the energy was measured with a BPM as a horizontal plane oscillation instantaneous magnitude. (Courtesy of J. Garland)

Using the Mathematica framework, interfaced to EPICS via activeX and .NET controls, a program was designed to analyse the EPICS BPM data, that can work with either many turns (on EMMA) or many bunches (in ALICE). Fig. 3 gives an example of ALICE long train (first 120 bunches) during standard operation. The plots in Fig. 1 were obtained also with this program.

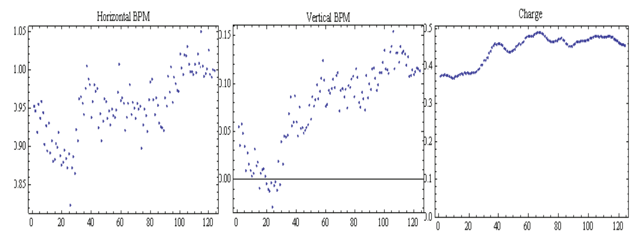


Fig. 3: Analysis of ALICE long train in Mathematica.

OUTLOOK

With VME/EPICS system and BPM programs getting matured, the latest EMMA runs have shown that some BPM cards need assessment. We plan to undertake a global card checking and, if necessary, adjust thresholds, clock circulator periods, clock delays, etc., using generator pulses instead of beam, and the Mathematica program which suits well a work of such kind.

With beam, a new AP with threshold switch will be tested.

ACKNOWLEDGMENT

The work is part of the BASROC/CONFORM/EMMA Project, and is supported by the UK Science and Technology Facilities Council.

REFERENCES

[1] S. Machida, "First Results from the EMMA Experiment", IPAC'11, San Sebastian, September 2011.

[2] A. Kalinin et al, "Diagnostic System Commissioning of the EMMA NS-FFAG Facility at Daresbury Laboratory", IPAC'10, Kyoto, August 2010.

[3] A. Kalinin, "Beam-Based Methods of BPM Center Offset and Resolution measurements", EPAC'02, Paris, June 2002. See also A. Kalinin, "Charge, Position, and Resolution Computing in EMMA BPMs", http://www.astec.ac.uk/emmafiles/meetings/tasks-meeting/mtg-43/

[4] I. W. Kirkman et al, "Calibration of the EMMA Beam Position Monitors: Position, Charge and Accuracy", MOPPR060, these proceedings.