FIRST TESTS WITH THE SIS18 DIGITAL BPM SYSTEM*

Andreas A. Galatis[1,2], K. Lang[1], P. Forck[1], A. Peters[1]
[1]GSI, Beam Diagnostics , Darmstadt, Germany
[2]TUD, Signal Processing Group, Darmstadt, Germany

Abstract

In this paper new approaches for BPM (Beam Position Monitor) measurements are described, which are needed in hadron accelerators with strongly varying beam parameters, such as intensity, accelerating frequency and bunch length. After the data collection and offline evaluation in 2005, first FPGA implementations of algorithms were completed in 2006 and tested at SIS18 and CERN PS. Main aspect of the first tests was the proof of concept in terms of online calculation feasibility. This includes online calculation of the needed integration windows as well as the baseline restoration algorithms. The realization of the hardware and the data handling are discussed. Least squares techniques were used for parametric fitting to gain bunch signal properties which can be used to monitor beam position.

PROBLEM DEFINITION

In most accelerators the exact position determination is crucial information for operation. At the SIS18 the desired resolution is 0.1mm. Compared to most accelerators [1, 2], the SIS18 has some peculiarities a BPM system needs to address, like the large frequency span of the RF from 850kHz to 5Mhz, the high signal dynamic, the injection of unbunched beams as well as the bunch length shortening of over one order of magnitude from some hundreds of ns down to 25ns FWHM [3, 4]. Therefore, novel techniques for beam position calculation which can manage all mentioned difficulties, have to be introduced. The demand of online position calculation introduces an extra challenge. The data is collected by an 125MSa/s-14bit-ADC [5] which is fast enough to overbear the alias boundary, considering the bandwidth of the pick-up units of 50MHz.

INTEGRATION WINDOWS

For the first online tests the algorithm as described in [3] was used. This method implements a filtering scheme which detects the flat regions between two successive bunches. It also includes a short averaging filter to keep the noise reduction algorithm simple in implementation. A length of five taps for the filter proved to be sufficient in most cases. The chosen median filter is easier to implement compared to a moving averaging filter, since it only requires relational operators to be evaluated.

* We acknowledge the support of the European Community-Research Infrastructure Action under the FP6 "Structuring the European Research Area" programme (DIRACsecondary-Beams, contract number 515873).

BASE LINE RESTORING

The method for removing the baseline (BLR) is addressed in [3]. It takes advantage of the known acceleration frequency as well as of the calculated integration windows. With that, two copies of the inverted original signal are used, each shifted by $+$ or $-$ $f_{RF}/2$. The stepwise mean of the those signals is added to the original signal, while the signal outside the previously calculated integration windows is neglected.

FPGA IMPLEMANTATION DETAILS

Using the stated median filter instead of a time averaging filter reduces the latency of calculation and the data overhead inside the FPGA. The median algorithm can be implemented as a parallel working network of comparators and multiplexors, which allows a latency of one clock after the last value arrives. In contrary to that an averaging filter would need one clock cycle for the addition of the last value as well as several clock cycles to produce the needed final division. After the input is flattened using the median filter, the comparison to determine bunch regions is done using parallel working shift registers, which again are optimised for latency minimisation.

The baseline restoration algorithm introduces a delay of half a RF period. The maximal delay is therefore defined by the largest bunch revolution time and is about $1.2\mu s$. For the calculation of the relation between Σ and Δ used in the calculation of the position a pipeline divider is used. The delay introduced by this element depends on the bit width one uses, in this case approximately $0.4\mu s$, leading to a maximum delay for the position calculation of about $1.6\mu s$.

CENTRAL DATA TREATMENT

The delay with which the individual bunch position reaches our central computer is variable. It heavily depends on the packet size one chooses for the ethernet frames, and therefore the number of data that can be sent inside it. It also depends on the used network protocol. These last two points are evaluated at the moment.

Position data of all 12 electronic boxes will be stored in a central point. This central computer has to manage a datastream of about 7MB/s per plane and PU at the maximum RF of about 5MHz. The total gross data will be about 170MB/s. This data has to be positioned correctly in real time and has to be stored. The used hardware for position calculation also provides several fast IOs, which can be used as Gigabit Ethernet Ports, to send the data to a central data storage device with the needed speed. Therefore, the system is going to be a server PC, which will build a dedicated Local Area Network (LAN) with the 12 Beam Position Processors and a Gigabit Ethernet Switch as junction. The acquired data can then be sent unidirectional from each PU to the server. Programming and controlling of the Beam Position Processors and data retrieving from the server can then be done over the normal Ethernet network.

FIRST TESTS AT CERN PS AND GSI SIS18

Tests where carried out at the GSI SIS18 in December 2006 using the algorithms in [3]. In parallel to those an implementation with the approach from [6, 7] was running. Both methods revealed problems when the signal strength is very low and the noise floor is relatively high. The approach from [6, 7] failed to syncronise due to the low signal strength, so position data could only be acquired using the GSI approach. The measurements implementing the GSI technique showed a standard deviation of the bunch-by-bunch position of about $0.23mm$.

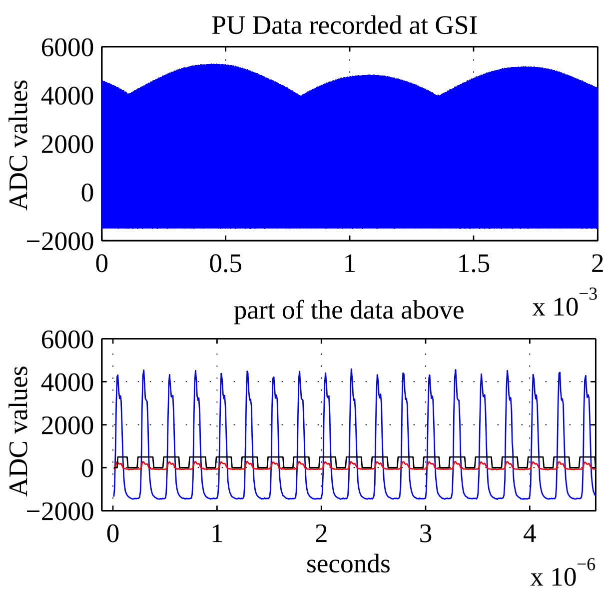

Figure 1: Top curve: complete signal recorder, bottom curve: zoom with sum (blue), difference (red) and window (black) signals

In Figure 1 top curve the complete recording with fluctuations of the peak in the sum signal can be seen. In the bottom curve a zoom into that data with the corresponding difference and the generated gating window – before the BLR – is depicted. In Figure 2 the same signal is shown after the BLR. The first two bunches are cropped due to the initialisation of the BLR algorithm. Figure 3 shows the calculated position of each bunch on $h = 4$ over a span of 2000 turns. The corresponding standard deviation of the position on a bunch-by-bunch basis is as stated

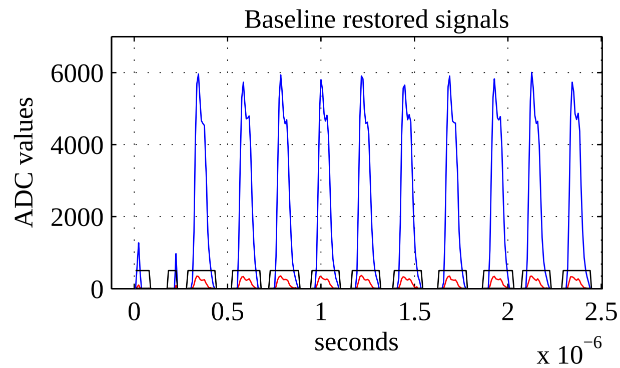

Figure 2: Sum (blue), difference (red) and gating window (black) signals after BLR

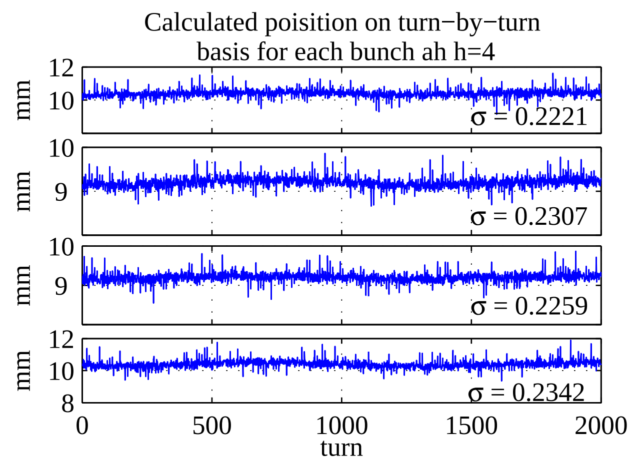

Figure 3: Bunch position on bunch-by-bunch basis on h=4

before about $0.22mm$.

Online tests carried out at the CERN PS using both implementations showed interesting results. While the CERN approach worked faultless, the GSI algorithm performed well in regions with good bunch spacing; a few errors occurring in regions where bunch splitting takes place. Nevertheless this error can be resolved if timing signals indicating a change of the harmonic are provided. The implemented BLR algorithm was not ideally set up for the kind of baseline the PS produces, resulting in a poorly restored signal. The integration window construction worked well except for some initialization faults on the first acquired bunch. This was corrected later off-line. The standard deviation of the acquired bunch-by-bunch position data of both approaches was about $0.27mm$.

LEAST SQUARES APPROACH

Expressing the physical properties of the PU signal by a mathematical model and fitting the acquired data to it, can give new approaches for removing the baseline and create the integration windows.

Parametric Modeling

The following nonlinear equation was used to model the four bunches which are instantaneously inside the acceleration ring:

$$f(n) = \sum_{k=1}^{4} (a_k e^{-\frac{(n-t_k)^2}{\sigma_k^2}} + y_k(n)) + e(n). \qquad (1)$$

$$y_k(n) = b_k n + r_k. \qquad (2)$$

In Equation 2 b_k is the slope of the baseline, r_k is the constant part of the displacement, $e(n)$ in Equation 1 is the noise term. The parameter vector $\boldsymbol{\theta}_k = [a_k\ b_k\ \sigma_k\ t_k\ r_k]^T$ is introduced, where a_k is the amplitude of the Gauss curve, σ_k its width and t_k its displacement inside the observation window.

Nonlinear Least Squares

The fitting parameters for a set of about 9000 consecutive bunches from a real data set recorded at SIS18 where calculated. Most of the results needed less than 13 iterations to provide stable results. In Figure 4 bottom curve the residuals of the fitting can be seen.

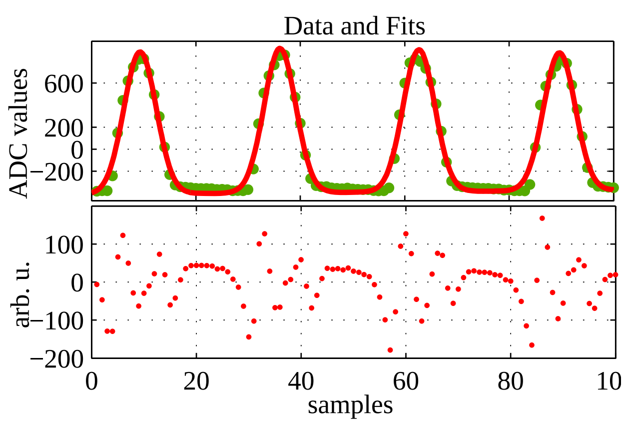

Figure 4: Top curve: original data (dots), fitted data (line), bottom curve: residuals

Results using fitted data

The position calculated from the real data recorded can be seen in Figure 5(a). Comparing the results from real and fitted data as seen in Figure5(a) and (b), gives a measure for the quality of the used model. The difference of both curves, plotted in 5(c), reveals that the inserted error is in the order of 10^{-5}mm with an approximately Gaussian distribution [8].

The implementation of nonlinear least squares (NLS) techniques on a FPGA is difficult, methods minimising the computational cost have to be considered. There are techniques solving the NLS techniques using iterative, minimal cost methods such as the Newton-Raphson (NR) technique. This method uses the previous estimate of θ as well as the Jacobian and the Hessian of the function minimising the NLS error criterion, to calculate the actual θ.

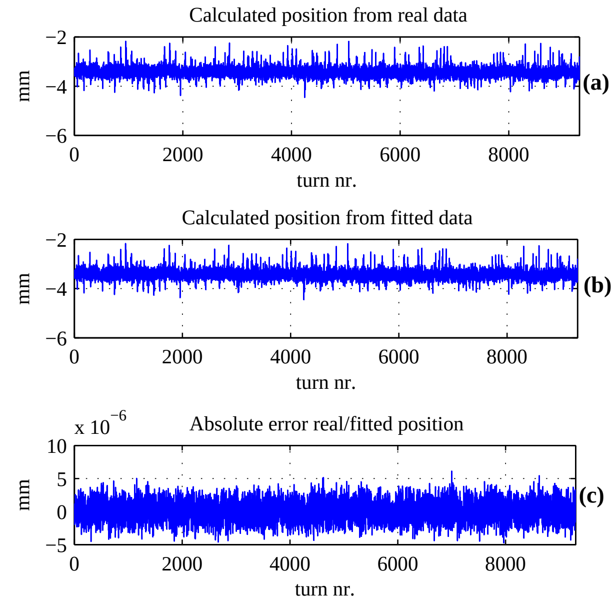

Figure 5: Calculated positions from real (a) and fitted (b) data for the second bunch on turn-by-turn basis, Δ from (a) and (b) in (c)

CONCLUSION

The first online tests using the methods introduced in [3] where successful on both the SIS18 and the CERN PS. The difficulties introduced by bunch gymnastics and varying baselines, can be solved using different approaches and have to be looked into more thoroughly.

The approach implementing the NLS can be used to remove the baseline part of the signal and allow for faster techniques for the integration window generation. More results on that topic are addressed in [8]. The computational complexity can be deducted from the number of multiplications and additions needed in the above update equation. The Hessian and Jacobian matrices have to be calculated at each iteration, nevertheless they are sparse and contain many double entries, reducing the computational cost.

REFERENCES

[1] G. Cancelo et al, "Digital Signal Processing the Tevatron BPM Signals", in Proc. of PAC '05

[2] M. Dehler et al, "New digital BPM System for the Swiss Light Source", in Proc. DIPAC '99

[3] A. Galatis et al, "Digital Techniques in BPM Measurements at GSI-SIS", in Proc. of EPAC '06

[4] A. Galatis et al, "Beam Position Monitor System Development for FAIR", GSI Scientific Report 2005

[5] Instrumentation Technologies, "http://www.i-tech.si"

[6] J. Belleman, "A New Trajectory Measurement System for the CERN Proton Synchrotron", in Proc. DIPAC '05

[7] Grzegorz Kasprowicz et al, "Digital Beam Trajectory and Orbit System, for the CERN PS", in Proc. of DIPAC' 07

[8] A. Galatis et al, "Beam parameter estimation for beam position measurements at the SIS18 accelerator", in Proc. of SSP' 07