DIGITAL TECHNIQUES IN BPM MEASUREMENTS AT GSI-SIS

A. Galatis* [1, 2], P. Kowina[1], K. Lang[1], A. Peters[1]
[1] Gesellschaft für Schwerionenforschung, Beam Diagnostics , Darmstadt, Germany
[2] Technical University Darmstadt, Institute of Telecommunications, Signal Processing Group, Darmstadt, Germany

Abstract

In this paper we describe new approaches for BPM (Beam Position Monitor) measurements, dedicated to hadron accelerators which have strongly varying beam parameters, such as intensity, accelerating frequency and bunch length. Following the adjustment of the signal dynamic, direct digitalization and treatment of digitized data, we should reach a BPM resolution of 0.1mm. Interchangeability between accelerators should be provided, which results in almost autonomous data treatment algorithms, free of external status and timing signalling. This should ensure the usability of the system in other bunched accelerator rings. Different operation modes are intended, allowing online storage of beam position data over full acceleration cycles as well as storage of beam waveforms in regions of acceleration that are of special interest e.g. transition, kicking, bunch gymnastics. First results of realised hardware/software combinations will be described and discussed.

PROBLEM STATEMENT

The present GSI Heavy Ion Synchrotron (SIS-18) lacks a reliable BPM system for bunch-by-bunch measurement, offering feedback capabilities. In preparation to the planned SIS upgrade and with the plans for FAIR [1] in mind a new BPM system has to be introduced. The design parameters of the existing and the intended facility imply that the use of commercial solutions for BPM devices cannot be used. Four attributes that undermine that fact are:

1. The large and fast changing radio frequency span from 850kHz up to 5MHz while acceleration cycle times are less then 1.5s.
2. The beam is bunched during acceleration. The length of the bunches varies from 1μs down to 50ns in one acceleration cycle.
3. The beam performs phase movements inside the RF buckets of such order that analogue PLL circuits cannot handle this.
4. The expected signal dynamics considering all expectations for the FAIR project, will be around 120dB [2]. This leads to strongly varying S/N ratios which have to be handled individually.

All these attributes led to the conclusion that for the evaluation of the digitized data new approaches should be made. One of the fundamental problems is the gate generation over which the individual bunch signal is integrated. Due to the phase shifting inside the bucket and the varying signal-to-noise ratio algorithms were introduced which generate the gating signal from the digitized signal itself, using almost no external parameters for timing.

The algorithm implementation is thought of running on dedicated hardware for every pick up. The hardware will consist of an analogue part handling the signal dynamics for ADC input adjustment and protection [3]. The ADCs themselves are capable of digitising at 14 bit resolution, while sampling at 125MSa/s. After the digitisation we have a Xilinx Virtex II Pro FPGA running at speed step 6, capable of handling integer values. This FPGA and all the communication in and around it are being controlled by a SBC (Single Board Computer) on the same hardware platform. In order to transfer data we have the option of a Gigabit-Ethernet connection or eight SFP (Small Formfactor Pluggable) connections for full duplex serial communication up to 1.25Gbit/s, which could be used for feedback control. 256MB DDRAM are available onboard, which can be used either as a buffer for storage off board or as a memory for data treatment [4].

PROPOSED SOLUTIONS - ALGORITHMS

To calculate beam position two important parameters have to be well defined:

1. Gate length and phase in respect to the bunch have to evolve in respect to the actual bunch length.
2. For correct integration over the length of a bunch, the floating baseline of the signal has to be restored.

As stated in the introduction the FPGA is capable to handle only integers hence the actual position calculation will be computed outside the FPGA.

Gate Construction

Two bunches in sequence ideally are separated by a data sequence of zeros, if one neglects the noise. In addition, we have a so-called baseline share in the signal. In order to distinguish between two bunches first the existing noise has to be suppressed.

To keep the noise reduction algorithm simple in implementation we decided to use an averaging filter with a small number of taps/elements. In most cases a length of five taps proved to be sufficient. A median filter was chosen, due to its easier implementation compared to a moving averaging filter. However using such a mean filter one would have to compute one addition and one division to produce a rounded integer value at each step, introducing a rounding error. The median only requires sorting of input data according to their magnitude, of which one uses the element in the middle of the constructed array.

For the gate generation we consider the fact that the regions between two bunches should approximately be linear straights while the bunches themselves can be approximated by Gaussian functions.

Ideally a stepwise accumulation would lead to a saw tooth like signal. The accumulation is computed via $x(i) = x(i-1) \pm y(i)$, where x is the accumulated and y one of the PU (Pick Up) signals.

Observing the bottom graphs of figures 1 and 2 we can see that the principle tendency of the accumulated signal is saw tooth like. Nevertheless we can observe a slope of the signal, which results from imperfect analogue signal paths, since the signal itself should be bipolar and AC coupled. This is the reason why it is indifferent if we accumulate by subtraction or addition in the above formula.

In figure 3 we see how the filtering alters the output signal, the blue curve showing the filtered signal from the bottom curve in figure 2. For the estimation of the beginning and end of the gating windows in both cases – whether at the beginning of the acceleration cycle as seen in figure 1 or as in a later time as in figure 2 – we look for the areas where the accumulated signal is “flat” over some samples. Flat is in this case defined as the amplitude change of ± 1 digit. Considering the shortening bunch length during an acceleration cycle, the coupled increasing of the RF frequency, as well as the sampling speed we use, we found that a filter length of 5 samples is the minimum working length.

The interpretation of the result of that method is relatively difficult if we observe the time during the bunching process and directly after that, due to the insufficient isolation between two subsequent bunches and/or the low signal to noise ratio. To better understand that we shall take a look at figures 1 and 2 bottom both curves are being generated using a 5 element long filter. In figure 2 the result is almost as expected, where as in figure 1 the slope is very steep. Nevertheless, the algorithm still produces valid integration windows.

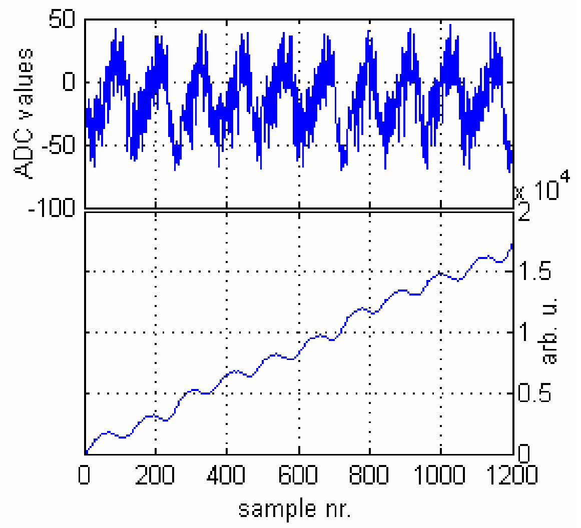

Figure 1: Top curve: original PU signal, bottom curve: accumulated and filtered PU signal.

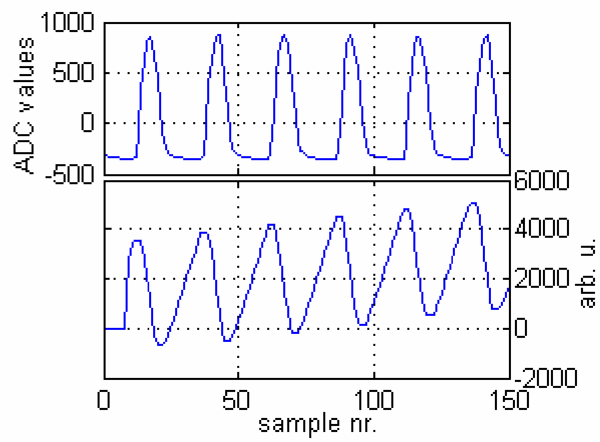

Figure 2: Top curve: original PU signal, bottom curve: accumulated and filtered PU signal

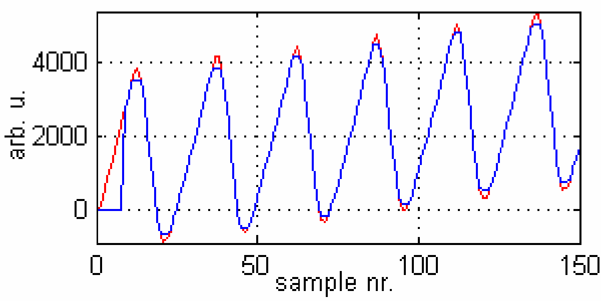

Figure 3: Red curve: accumulated PU signal, blue curve: accumulated and filtered PU signal.

If we observe in figure 3 the blue curve we see at the beginning the time the filter needs to initialize, but can also observe the flat regions where the red curve is visible, which finally give us the gating window.

Base Line Restoring

For precise restoring of the baseline shift caused by the DC blocking of the amplifier circuitry, one should fit an exponential curve to each individual bunch. This procedure would be rather heavy to compute, even if we used a small number of parameters.

Alternatively, knowing the acceleration frequency, neither the amplitude or the phase are of interest and irrespective of where we gain it from, and having obtained the gating windows, we use two copies of the original signal, one which is delayed by half a RF period and a second one which is delayed by a whole RF period. The last one as well as the original one is being inverted. We then calculate the stepwise mean of the inverted signals; afterwards we add to that the signal which had a delay of only half a RF period. We can sum that up in:

$y(i) = (-x(i) - x(i-f))/2 + x(i-f/2)$; f representing the frequency in samples.

This method removes the DC component and some very low frequency signal parts, and due to its simplicity it is easy to implement.

Using a part of the signal data from figure 2 top curve we obtain:

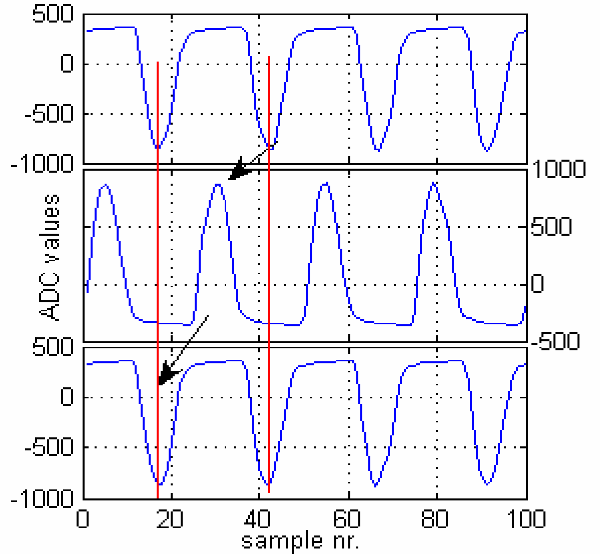

Figure: 4 Top curve: inverted original signal, middle curve: original signal delayed by half an RF period, bottom curve: inverted original signal delayed by one RF period. The black arrows show the position of one particular bunch; the red lines mark the area of interest.

Computing the previous formula would result in a sinusoidal signal. The regions between two consecutive bunches would be suppressed as intended, but we would also have the inverted and delayed signal on the negative y-axis. We process the part of the signal in between the red borders shown in figure 4 and inside the respective integration window. In figure 5 we see the results of that approach.

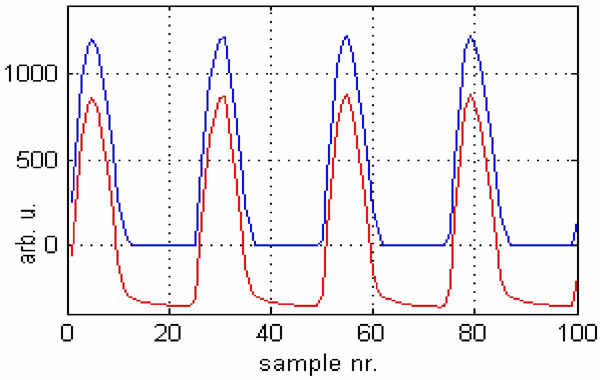

Figure 5: Blue curve: signal after baseline restoring, red curve: original PU signal.

In figure 5 we can see a good restoring approach at the starting point of the bunch, whereas at the end we loose some part of the tail. We have tested this method of BLR (Base Line Restoring) algorithm on real data sets even on these signals with poor S/N ratio and have found it to work in principle, the error introduced has to be calculated. At acceleration cycle stages as shown in figure 5 we could substitute the introduced BLR algorithm by fitting a straight curve into the starting and the ending point of a bunch. On data from the beginning of acceleration as shown in figure 1 a straight line approach fails, the introduced error is dominated by the poor S/N ratio.

Other methods which could be used where introduced in [5, 6].

Results

Using signals like the ones in figure 2 top curve we obtained the curves in figure 6. The red curve shows the estimated position using the introduced BLR method, the blue one corresponds to linear BLR. We see that the tendency is the same in both curves; the variation is visibly smaller while using the enhanced BLR. The standard deviation is 0.28mm and 0.14mm respectively.

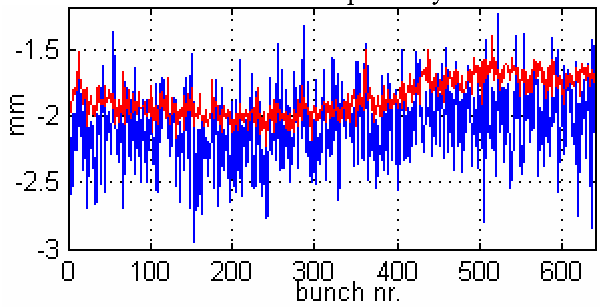

Figure 6: Calculated position over approx. 150 turns on harmonic 4, blue curve showing the results with linear BLR, the red curve showing results using the introduced BLR.

The hardware and its core software reached their first complete implementation and realization in June 2006 [4, 7]. The software that implements some of the introduced methods should be imported into the hardware in the next months. First measurements and online evaluation of the proposed can be expected shortly after that.

ACKNOWLEDGMENTS

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).

REFERENCES

[1] A. Peters et al, "Beam Diagnostics Challenges in the FAIR Project at GSI", BIW 06.
[2] P. Kowina et al, these proceedings.
[3] J. Schölles et al, DIPAC '05.
[4] www.i-tech.si
[5] A. Galatis, A. Peters, "Digital Beam Position Measurement at GSI-SIS and CERN-PS", DIPAC '05, Lyon, June 2005, see: http://www.jacow.org
[6] J. Belleman, "A New Trajectory Measurement System for the CERN Proton Synchrotron", DIPAC '05, Lyon, June 2005, see: http://www.jacow.org
[7] Collaboration developments of: "DIRACsecondary-Beams, Task 21: SIS100_5".