BPM SEARCH ALGORITHMS FOR BEAM INJECTION AND EXTRACTION AT FERMILAB

C. Briegel[1]
[1]*Fermi National Accelerator Laboratory, Batavia, Illinois, USA*

ABSTRACT

The requirement of correctly returning the beam position on the first extraction cycle from Main Injector to NUMI eliminated using a time-triggered based system. Instead, the BPM was triggered by the kicker and several algorithms were written to find the beam position based on the beam intensity measurement. The BPM position is used to auto-tune the beam line to the NUMI target producing neutrinos for MINOS in Minnesota.

NUMI beam consists of 1-6 batches at similar intensities and each search strategy has advantages and disadvantages. A set of algorithms provide peak, edge, and max-edge search for both injected and extracted beam.

This paper will discuss the various algorithms implemented and modifications starting with test signals, first beam pulse, and further improvements for machine operation. The system was implemented for NUMI and currently being installed in six transfer lines. The same software co-exists with Recycler BPMs and the test-bed for Main Injector BPMs.

SYSTEM OVERVIEW

The requirement for returning valid positions and intensities on the first beam pulse extracted from MI(Main Injector) to NUMI(Neutrinos at the Main Injector) excluded a time-triggered based system. The resulting implementation was directly applicable to injection and extraction beam lines. Thus, it is being used for the Rapid Transfer and MI Abort beam lines at Fermilab. The software evolved from the Recycler Ring BPM project and one of the evolutionary steps was to enable a call-back routine for each specified filter. The call-back routine could then implement a search algorithm to find qualified beam intensity and position of individual batches (a batch is 84 53MHz bunches in 1.6 us.). The same software and similar hardware persists in Recycler, NUMI beam lines, Rapid Transfer beam lines, and the MI abort line. Further, this system has been used as an orbit verifier prior to beam extraction to NUMI, beam line turner, and testing MI BPM functionality with minimal changes to the software.

The NUMI BPMs are located in three locations; MI60S, MI60N, and MI65. There are 23 capacitive pickup BPMs and 4 strip-line BPMs which see circulating beam as well as the last turn extracted to NUMI. Figure 1 displays the last 20 turns prior to extraction down the NUMI beam line for a MI60S BPM located in the MI ring. The implementation attempted to optimize the time required to find the last turn in 1024 data points which encompasses approximately 40 turns in MI or 440 usecs.

Hardware

The front-end hardware is based on a common architecture found at Fermilab. It consists of a VME crate, a MVME24xx processor, and a Technobox PMC(PCI Mezzanine Card) loaded with FPGA(Field Programmable Gate Array) firmware to decode Tevatron clock events. Additional functionality specific to the BPM is provided by an Echotek ECDR-GC814 digitizer, an Echotek PMC trigger card, and an IP(Industry Pack) carrier populated with Fermilab-designed IP cards loaded with FPGA firmware to provide Tevatron clock events, Beam Sync clock events, MDAT(machine data), and delay timers.

Each Echotek digitizer contains 8 ADCs with 14-bit resolution capable 80 MHz conversions. Each ADC channel has 4 Graychip GC4016 digitial receiver channels. A single Graychip receiver channel contains a NCO(numerically controlled oscillator) mixer with .02 Hz tuning resolution, a 5-stage CIC(cascaded integrator comb) filter, one programmable 21 tap FIR(finite impulse response) filter, one programmable 63 tap FIR filter, and one 256 tap resampler FIR filter. The data collection is armed by either a clock event or periodically and triggered with programmable delays via a beam signal from the beam sync clock or immediately. The board provides 128K 32-bit samples of FIFO(version 1) or dual port ram(version 2) for captured data. The board can be triggered multiple times with a specified burst count for each trigger.

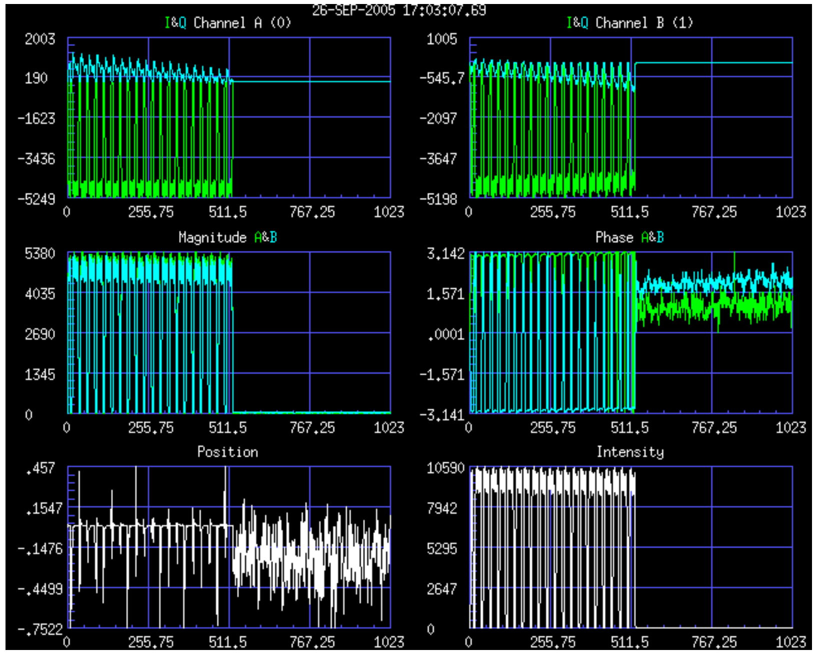

Figure 1: 20 turns before last turn

Software

The software is implemented in the context of a Wind River vxWorks operating system and a Fermilab MOOC(minimal object oriented communications) skeleton which provides data to console applications. The software is layered with a BPM class, a TRIGGER class, an ADC class, and an Echotek class written in C++. The BPM class interfaces to the ADC class which hides the details of the implemented hardware. The ADC class interfaces to one of two Echotek classes for two different versions of hardware. The Echotek class is implemented as a vxWorks driver with hooks for manipulating the hardware directly.

The system provides 16 user specifications which arm and trigger the data collection. The conditions are qualified based on **measurement type (**RepetitiveSingleGate, OneShotMultipleGate, OneShotSingleGate, RepetitiveNoGate, TestMeasurement, Prearm, TimingScan, TurnToTurnPeriod), **arm event** (ArmAutomatic, ArmTclkEvent), **trigger event** (TriggerPeriodic, Trigger External, TriggerAutomatic, TriggerBeamSyncEvent), **beam mode** (Antiproton, Proton, Calibration), and **beam type** (InjectExtract, Hot, HotHead, HotTail, Cold, ColdHead, ColdTail, 53MHzStacking, 53MHz, 2.5MHz, 53MHzNarrowBand, 2.5MHzNarrowBand, 53MHzWideBand, 2.5MHzWideBand). Each of these 16 specifications is coupled with a Graychip filter specification and a possible search algorithm.

Position and Intensity Evaluation

The output of the Graychip is an I (in-phase) and a Q (quadrature-phase) term. The I and Q pair determine the amplitude and phase of the pickup. An ADC channel is used for a BPM pickup and two pickups comprise a BPM. The intensity and position is calculated for A and B plates as follows:

$$A = B = \sqrt{I^2 + Q^2}$$

$$\text{Intensity} = A+B$$

$$\text{Position} = (A-B)/(A+B)$$

SEARCH ALGORITHMS

The current set of search algorithms are FilterLtEdge, FilterRtEdge, FilterRtMxEdge, FilterLtPeak, FilterRtPeak, FilterMaximum, and FilterTimed where the Lt implies a left-to-right search of the data versus Rt implies a right-to-left search of the data.

The goal was to minimize the data evaluated and provide a robust solution for varying beam conditions. The implementation consists of a common qualifier to eliminate noise from generating erroneous beam position and several algorithms to find batches of maximum intensity which can be used to as a best-measure of the beam position.

Beam Qualifier

The data is sampled and intensity is calculated from the data samples. The threshold is set to be the averaged intensity plus a programmable rate of rise from the previous reading plus a fixed offset. This resulting threshold has a maximum value. If the sample is above the threshold, then the algorithm starts a search for intensities which imply valid positions of individual batches.

Peak Search

The peak search utilizes a sliding window of a programmable dimension such that the window fits over the intensity waveform. The peak is determined by a maximum value in the centre of the sliding window. The intensity plot in Figure 2 shows nicely peaked values for 5 batches of beam for NUMI. While NUMI can have 6 batches, the left-most batch is always referred as batch one. The spacing of batches as defined by the edge search can be used to denote missing batches after batch one, but there is no way of knowing the true batch number unless all six are available.

The peak search is currently the preferred search algorithm for NUMI.

Edge Search

The edge search utilizes a fixed offset from the first peak. The peak is determined by a sliding window of a programmable dimension such that a new value is no longer larger than the average inside the window. A variation is to simply find the value which is greater than its successor. A table is used with fixed offsets from this edge. The intensities are qualified with respect to the original threshold to determine if batches are missing. Then, the batches are shifted such that the first batch is qualified.

Timed Search

A timed search is used to find either a test signal or real beam with a set of fixed offsets into the data collection. The signal must be timed by adjusting either the Echotek delays or trigger delays to the digitizer. The intensities are then qualified in post-analysis.

Maximum Search

In searching for Rapid Transfer beam, the signals contained spikes and potentially low signal-to-noise ratio. Fortunately, there is only one batch of beam for Rapid Transfer and time between pulses. This search algorithm simply analyses the entire data set and simply takes the maximum intensity as its single valid position.

Post Analysis

As an option, a post-analysis is done to further qualify the intensity. All batches must be within a programmable percentage of the maximum intensity. All subsequent batches must be within a programmable percentage of the average. The assumption is the intensity should be of a similar magnitude. This feature was implemented to eliminate a beam halo left in the machine from a previous cycle which was not adequately aborted. Since the positional average was used to auto-tune the line, the halo had an undesirable result. An additional feature of de-selecting batches for the average was implemented.

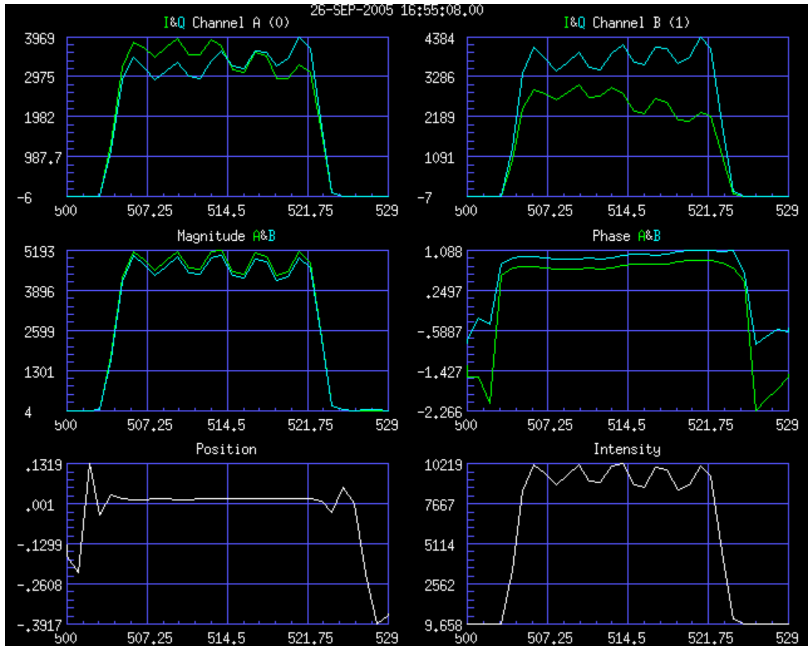

Figure 2: View of last turn for NUMI 5 batches; I, Q, Position, & Intensity

SUMMARY

The NUMI BPMs provided a beam position on the first transfer and aided in tuning the beam line such that by the tenth pulse beam hit the near target at Fermilab. Subsequent implementations have been achieved with minimal modifications to the software. Further, the software has maintained backwards compatibility with extensions for future implementations and testing.

Since the initial implementation, the qualifying threshold was modified to accept low intensity beam, the peak search is preferred over the edge search algorithm, the post-analysis was added to eliminate a halo of beam inadequately aborted, and the maximum search algorithm was added to eliminate noise spikes for single batches.

I would like to recognize Peter Prieto, Duane Voy, and Craig McClure for their considerable work towards this implementation. My efforts were focused primarily on the ADC class and Echotek class which is where the search algorithms resided.

REFERENCES

[1] R. Webber, J. Crisp, P. Prieto, D. Voy, C. Briegel, C. McClure, R. West, S. Pordes, M. Mengel, "Fermilab Recycler Ring BPM Upgrade Based on Digital Receiver Technology", Fermi National Accelerator Laboratory, Batavia, IL. 60501-0500, USA

[2] "ECDR-GC814-FERMI USER MANUAL", Echotek Corporation, Huntsville, AL, 35816

[3] "ECDR-GC814-FV2 USER MANUAL", Echotek Corporation, Huntsville, AL, 35816