

PYTHON BASED APPLICATION FOR BEAM CURRENT TRANSFORMER SIGNAL ANALYSIS*

M.C. Paniccia[†], D.M. Gassner, A. Marusic, A. Sukhanov,
Brookhaven National Laboratory, Upton, USA

Abstract

There are a variety of beam current transformers that are used at all accelerator facilities for current and bunch charge measurements. Transformer signals are traditionally measured using integrator electronics followed by a digitizer. However, integrator circuits have a limited bandwidth and are susceptible to noise. By directly digitizing the output of the transformer, the signal bandwidth is limited only by the transformer characteristics and the digitizing platform. Digital integration and filtering can then easily be applied to reduce noise resulting in an overall improvement of the beam parameter measurements. This paper describes a python-based application that performs the filtering and integration of a current transformer pulse that has been directly digitized by an oscilloscope.

INTRODUCTION

Current transformers have been the standard instrument for current and bunch charge measurements across all accelerator facilities. At the BNL Low Energy RHIC electron Cooling (LEReC) facility [1], a Bergoz Integrating Current Transformer (ICT) is utilized for calibrated bunch charge measurements. The LEReC beam contains a complex bunch structure defined by the 704 MHz fiber laser producing individual electron bunches of about 40 ps full length which are grouped together into a single macro-bunch. The macrobunches contain thirty individual bunches and macrobunch trains are separated at ~ 9 MHz frequency, as shown in Fig. 1.

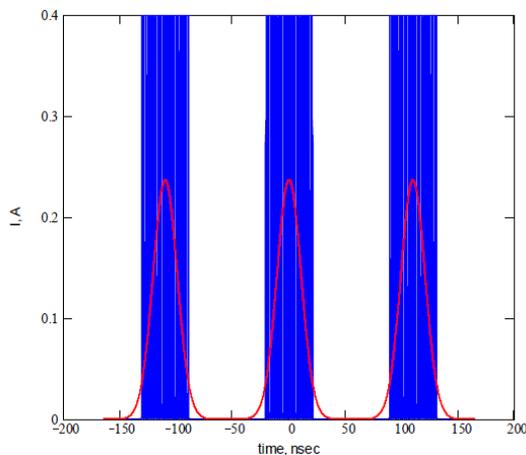


Figure 1: The LEReC beam structure. Thirty electron bunches (blue) spaced by 1.4ns placed on a single ion bunch (red), with ion bunch repetition frequency of 9 MHz.

* Work supported by the US Department of Energy under contract No. DE-AC02-98CH10886.

[†] mpaniccia@bnl.gov

The ICT provides precise macrobunch measurements down to the pC level and provides a pulse proportional to the bunch charge with a fixed width of approximately 70 ns [2]. Initial configuration of the ICT measurement system used a Bergoz Beam Charge Monitor Integrate/Hold/Reset (BCM-IHR) integrator module. When the LEReC program transitioned to long macrobunch trains and continuous wave (CW) beams, the integrator could no longer be used to calculate beam charge. Therefore, to utilize the ICT during these operational modes, the output of the ICT is directly digitized with an oscilloscope. A custom Python application integrates each pulse to provide bunch-by-bunch charge calculation and estimated beam current.

HARDWARE SETUP

The ICT is a specially designed current transformer that integrates the charge of the passing beam pulse and then outputs the results. The immediate integration allows for the measurement of beam pulses with picosecond pulse widths. In addition, it reduces the effects of pulse-to-pulse crosstalk and eddy-current loss [3]. At LEReC, the ICT is an in-flange module with 5:1 turn ratio and is located in the Injection section of the LEReC accelerator beam line [4]. The output is carried over LDF1-50a, 1/4" Heliax cable to the monitoring electronics 100 meters away. A SPDT relay is used to connect the ICT to either the BCM-IHR electronics or the Keysight DSO-X-3024A oscilloscope and can be controlled remotely. Controls for the BCM-IHR gain settings and calibration settings are remotely available using a VME-based digital output module and the output is digitized on a custom Zynq-based digitizer chassis. Both the oscilloscope and the BCM-IHR electronics run off an external 1 Hz trigger.

BCM-IHR vs. Oscilloscope

Macrobunch charge is determined by integrating the output pulse from the ICT. Integration in the BCM-IHR electronics module is performed in two stages: first, a positive integration window, followed by a negative integration window. The two windows are summed together and provide a DC level proportional to the charge captured. The two-window configuration performs a baseline subtraction that reduces the effects of baseline noise and baseline droop. Integration window length can be adjusted using a potentiometer and is limited to 0.1 - 7 μ s; both windows have the same duration. The module can operate at a maximum repetition rate of 10 KHz [5]. At LEReC, the integration windows are set to 7 μ s in order to capture up to 60 macrobunches. However, the BCM-IHR integrator is limited to 40 nC before saturating, limiting the number of macrobunches to 10 at nominal individual electron bunch charges of 130 pC.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

In order to measure longer pulse trains with higher electron bunch charges, the ICT output is switched to the oscilloscope. The oscilloscope captures the raw output from the ICT and transfers the data over the network. A simple Python application provides remote control for the oscilloscope configuration (time base, vertical range, etc.) and can be configured to provide a 20 – 200 μ s capture window before the sampling rate decreases. The application then integrates each macrobunch individually and provides an average bunch charge, estimated beam current, and an array showing the bunch-by-bunch variation.

PYTHON APPLICATION

The Python application utilizes the BNL custom Python libraries designed to interface with the RHIC Controls system [6]. The application is split between a manager used for continuous logging and a GUI used for analysis.

Application Manager

Collection Method Communication with the oscilloscope is done with a custom socket library developed by the Keysight application team and the oscilloscope is controlled with SCPI commands. Upon starting the application, the oscilloscope is initialized by resetting to factory defaults, clearing any existing errors, and configuring the horizontal, vertical, and trigger settings. The application allows the user to configure the vertical and horizontal configurations at any time during operations. After configuration is complete the oscilloscope is armed for single capture and after the scope is triggered the data is transferred as 8-bit binary values. Using the Python struct package the hexadecimal values are unpacked into a NumPy array of double-precision values. At the default window size of 20 μ s, 80,000 samples are transferred in < 1 second.

Noise Filtering Ground loops, switching power supplies, RF cavities, and vacuum pumps can all add noise into the measurement system. This noise is present on the baseline and can be as large as the ICT output for small charges. The typical noise profile for the LEReC ICT consists of high frequency burst reoccurring at a low frequency out of phase with the ICT output. The bursts of high frequency are within the bandwidth of the ICT output making it difficult to filter out using the standard techniques. To remove both the low frequency and high frequency noise and preserve the integrity of the ICT output, the application employs a wavelet denoising technique using the pywavelt Python library.

Wavelet denoising uses the Discrete Wavelet Transform (DWT) and applies filters directly to discrete levels. The DWT decomposition filters the signal into several discrete levels containing the time-frequency relationship over a certain frequency range. As the levels increase, the center of the frequency range increases. Low frequency baseline ripples are removed by zeroing out the first level(s) of decomposition. Since the ICT output spectrum consists of mostly high frequencies, the low frequency zeroing has little impact on the signal of interest. At the higher frequencies, the filter removes high frequency content that is lower

than a specified threshold. Pulses from the ICT are well above the threshold and remain unaffected. Additionally, this process smooths out quantization steps for low resolution pulses, as shown in Fig. 2.

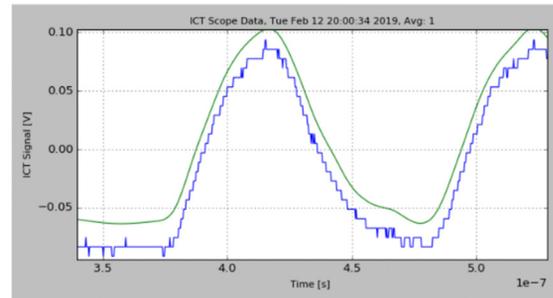


Figure 2: The LEReC ICT pulse for single macrobunch. The blue trace is the raw output from the oscilloscope and the green trace is the filter response.

The application provides the user options to turn on/off the wavelet filter as well as choose what levels to filter for both baseline filtering and high frequency filtering. Additionally, the user can select various wavelets to use for the DWT to optimize the filter performance. After the filtering is complete on the user specified levels, the ICT signal is reconstructed by performing an inverse DWT.

Pulse Binning After filtering, the individual pulses from the ICT are separated out into individual NumPy arrays using a binning process. The binning process searches through the signal array and slices out the pulses when identified. Traditional threshold detection can be achieved using vectorization,

```
index = numpy.nonzero( x > thresh )
```

or, more cumbersome, using a for-loop. However, threshold detection is susceptible to noise. Therefore, each individual pulse is picked out of the data array using slope detection. The output of the ICT has a fixed rise time of 40 ns and a fall-time of approximately 60 ns. The rise-time of the pulse is greater than the rise-time of the noise and can be easily distinguished after taking the first derivative. A built-in NumPy function, diff, performs the first derivative and returns an array of the results. A second function, index, from the peakutils library returns an array containing the index location of all the maxima values. Using the two functions, the start of each ICT pulse is captured in two lines:

```
diff_x = numpy.diff(x)
```

```
index = peakutils.index( diff_x ).
```

Integration Charge per pulse is proportional to the integral of each ICT pulse. Each bin, containing a single ICT pulse, is integrated using the.simps function from the SciPy integrate package, which performs a Simpson Integration. Prior to integration, any DC offset from baseline droop is removed. As defined above, macrobunches are 9.1MHz apart and are close to the cutoff frequency for the ICT, < 15 MHz. As a result, the ICT pulses never fully return to

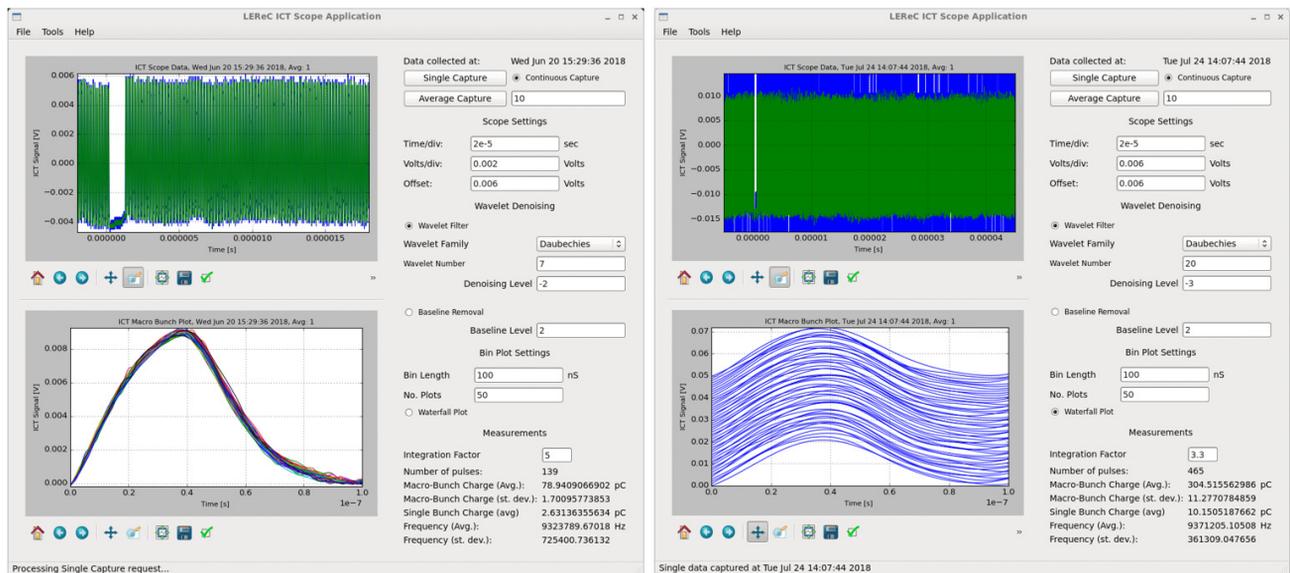


Figure 3: Example of a LEReC ICT GUI. In the picture on the left, the application displays each ICT pulse layered on top of each other and the right picture displays the ICT pulses in a waterfall type plot.

zero and cause the baseline to droop as the number macro-bunches increase. Eventually the baseline settles out to a value proportional to the beam current. To remove this droop, the baseline is approximated linearly from beginning to end of the ICT pulse in each bin and subtracted out.

After each pulse is integrated, the charge values are logged in an array. Average charge per macrobunch and the standard deviation is calculated and logged along with the frequency between macrobunches. In CW mode, the beam current is estimated by multiplying the average charge per bunch and the total number of macrobunches. The calculated beam current is compared to a DCCT to calibrate the ICT measurements. Additionally, a power spectrum of the bunch-by-bunch charge array using a simple FFT is provided.

Application Interface

A GUI was designed in the QT Interface Designer software to visualize the ICT pulse and the post-processing performed by the manager. The output of the Designer software is a user interface (UI) file. The Python PyQt library builds the GUI from the UI file, removing the need to manually layout the GUI in the Python script.

The GUI provides two charts displaying the full window response from the oscilloscope and a closeup of the bins. In the close-up chart, the bins can either be plotted on top of each other or as a waterfall to easily view variations between pulses. Data displayed on the GUI is independent from the device manager to allow the user to manipulate the data without corrupting logged information. The user can adjust filters, integration factors, and bin sizes to optimize the performance and then copy the changes into the manager. An example of the GUI can be seen in Fig. 3.

CONCLUSION

The Python application provides functionality to the LEReC ICT for all beam conditions and provides a simple user-friendly interface for analysis. The application can be easily adapted for different current transformers and different oscilloscopes for quick and reliable measurements. The application was implemented at the LEReC facilities for the 2018 run and was the primary resource for ICT measurements during the 2019 run.

REFERENCES

- [1] A. V. Fedotov *et al.*, “First Electron Cooling of Hadron Beams Using a Bunched Electron Beam”, presented at the NAPAC’19, Lansing, MI, USA, Sep. 2019, paper THZBA5.
- [2] *Integrated Current Transformer User’s Manual*, Bergoz Instrumentation, Saint Genis Pouilly, France, Jan. 2018, https://www.bergoz.com/sites/www.bergoz.com/files/ictmanual4-0_0
- [3] K. B. Unser, “Design and Preliminary Tests of a Beam Intensity Monitor for LEP”, in *Proc. PAC’89*, Chicago, IL, USA, Mar. 1989, pp. 71-74.
- [4] T. A. Miller *et al.*, “Overview of the Beam Instrumentation and Commissioning Results from the BNL Low Energy RHIC Electron Cooling Facility”, presented at the IBIC’19, Malmö, Sweden, Sep. 2019, paper MOBO01.
- [5] *Beam Charge Monitor Integrate-Hold-Reset User’s Manual*, Bergoz Instrumentation, Saint Genis Pouilly, France, Jan. 2018, <https://www.bergoz.com/sites/www.bergoz.com/files/bcm-ihrmanual3-0>
- [6] K. A. Brown *et al.*, “Python and MATLAB Interfaces to RHIC Controls Data”, in *Proc. ICALEPCS’17*, Barcelona, Spain, Oct. 2017, pp. 765-769. doi:10.18429/JACoW-ICALEPCS2017-TUPHA153