Beams Doc. 4932-V3

An ACNET Application Program to Measure the Energy Spread of Multi-turn Beam in the Booster at Injection

Jovan J. Nelson,* C. M. Bhat, B.S. Hendricks, and M.J. Yang

Accelerator Division, Fermilab

August 10, 2015

Abstract: We have developed an ACNET based console application program to be used to measure energy spread of injected proton beam from Linac, at the energy of 400 MeV. Depending on beam intensity requirement the injection process could take a number of Booster turns. As we launch the program it secures the control of a digitizing oscilloscope which has Booster Wall Current Monitor (WCM) signal at its input or warns if scope is being used by other users. It also provides user abilities to configure scope settings for optimal data acquisition, and to select a Booster beam event for the measurement. Subsequently, a special one-shot timeline is generated, with the approval from Main Control Room, to initiate the selected Booster beam event. An ACL script is used to read operational timeline before launching the 1-shot timeline, and to restore it afterwards. After the completion of injection from LINAC the script issues command to notch kicker to produce a gap in the injected beam. After the Booster event, this console program collects and analyses data to extract beam energy spread. We illustrate a case with an example.

Overview

The Booster is one of the oldest rapid cycling (at 15 Hz) proton synchrotron in the world, operating at 8 GeV extraction energy since May 1971 [1, 2]. Many improvements have been made over the past four decades to increase the beam intensity per Booster batch by injecting multi-turn beam in to the Booster and to increase its average beam power to the down-stream accelerators and to the low energy neutrino program. All these years the Booster was operated with <15 Hz for beam delivery — full potential of the Booster is yet to be realized. One of the important goals of the Fermilab future upgrade program, "Proton Improvement Plan" at Fermilab [3] is to extract the beam at 15 Hz rate from the Booster all the time. Long range plan of Fermilab [4] is to increase the Booster beam delivery cycle rate from 15 Hz to 20 Hz. Hence, the current Booster plays a very significant role in the near future of the Fermilab.

In support of the proposed upgrades to the Booster, a thorough investigation on the properties of the beam at the injection energy is extremely valuable; in particular measurements on beam energy spread. In the past, many attempts have been made to measure beam energy spread (e.g., ref. [5]) in bits and pieces, often only soon after major upgrades in the complex. However, till very recently no measurements were made on the multi-turn beam at injection. During 2013-2014, we developed a very robust method to measure the beam energy spread by using the wall current monitor data collected after a notch of width ≈ 40 nsec is formed in the injected beam [6, 7]. These attempts lead us to develop a console application program which would enable us to measure the beam energy spread at injection on request.

The Application Program

The Physics behind the Program:

As the beam circulates in a synchrotron, the higher and lower momentum particles drift differently relative to the synchronous particle. For example, at 400 MeV the particle with slightly more momentum will bend less in a pure dipole field relative to the synchronous particles at 400 MeV. Similarly, a lower momentum particle will bend more and travel more inside. A schematic view of beam particle distribution of such a beam injected from the Fermilab LINAC in to the Booster is shown in Fig. 1. If one creates a short gap (notch) in a

fully debunched beam using a fast beam kicker, the slipping of high and low momentum particles can be seen very clearly as shown in the last picture below. The WCM records the line-charge distribution of the beam particles as it circulates in the ring. The first trace of the WCM after formation of the notch will be essentially a continuous beam with a gap of width "Wnotch" and the subsequent traces represent evolution of notch filling as the beam particles continue to slip. By measuring the notch width and the time required for the highest momentum particles cross (or touch) the lowest momentum particles at the head of the beam, "Tgraze", we can measure beam energy spread using,

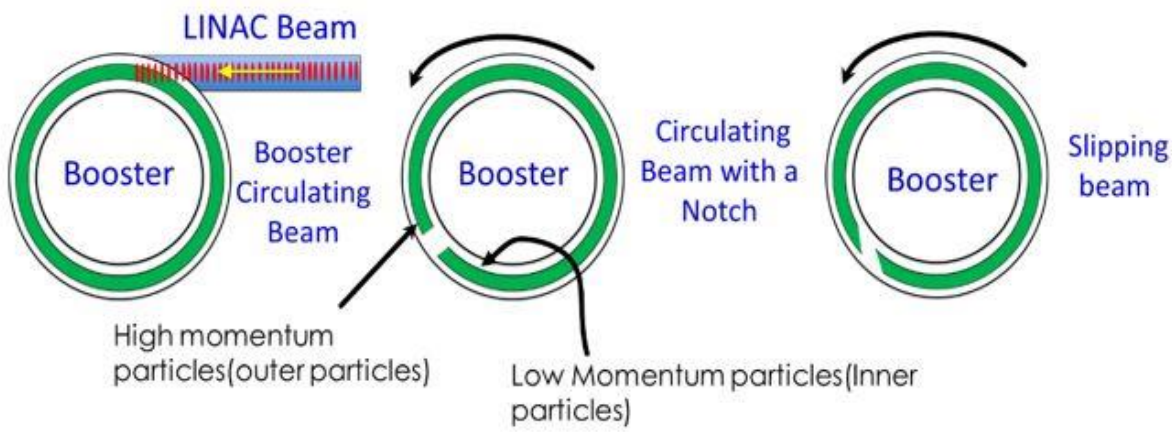

Figure 1: The beam pulses enter from the LINAC and create a continuous beam in the Booster (left). Notch at its creation (middle). High and low momentum particles are also shown. Beam after some time; indicating slipped beam and particles line up 'touch'(right).

$$\Delta E = \frac{\beta^2 Es}{|\eta|} \frac{Wnotch}{Tgraze}$$

where β is the relativistic speed, Es is the synchronous energy, and η is the slip factor, we can calculate the energy spread (ΔE).

Program Overview:

This application program was designed based on Fermilab ACNET Console Environment to

ensure its availability to user, and to have easy access to the hardware devices that are essential for the measurement. The WCM in the Booster ring and the Tektronix (TDS7154B Digital Phosphor Oscilloscope 1.5GHz 20GS/s) scope used here are shown in Fig. 2. This program controls

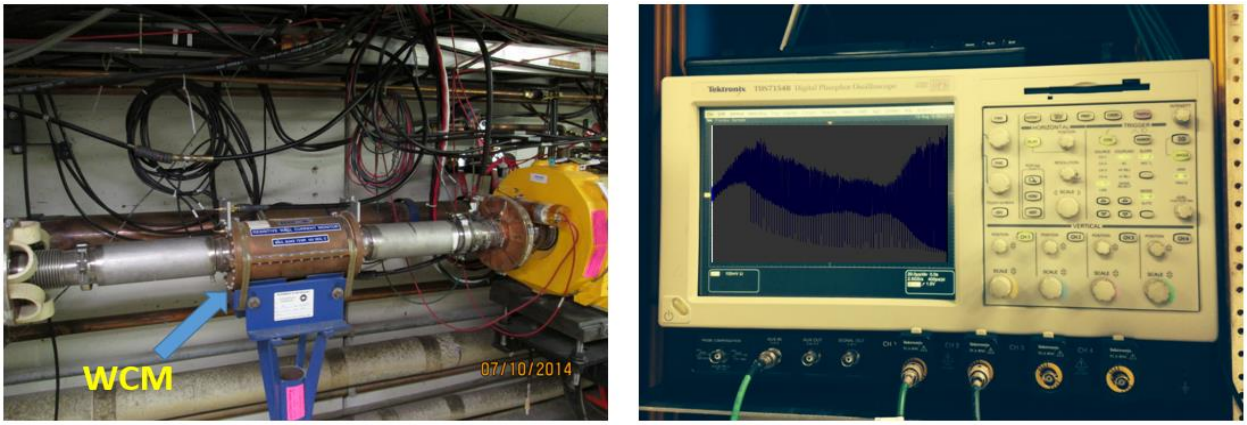

Figure 2: Wall Current Monitor in the Booster ring (left) and the TDS7154B Digital Phosphor Oscilloscope 1.5GHz 20GS/s scope used for data collection (right) are shown.

various types of hardware that are necessary to measure the energy spread of the beam. A schematic view of the block diagram representing the structure of the application program is shown in Fig. 3. The program works as follows:

1) Check if the executing the program is safe (all necessary instruments are available to measure the Booster energy spread) and warns if another user is using the same devices. If so one cannot proceed.

2) Sets necessary parameters on the wall current monitor (WCM) scope like, a) channel upper and lower amplitude levels (default values are 1 V and -.04 V, respectively), b) number of data points (default is 50000 for "Record length"), c) space between the data points ("interval" 0.4 nsec), d) TCLK event of interest.

3) Executes an ACL script upon permission from the Main Control Room (MCR) crew chief which enables a) a "1-shot" timeline with beam on $17 (a TCLK beam event), b) moves all timers related to RF paraphrase settings to a later time, c) moves notch timer to soon after the injection, d) moves radial feedback loops to later time so that one can

collect nice WCM data on the injected beam for first 200 micro seconds.

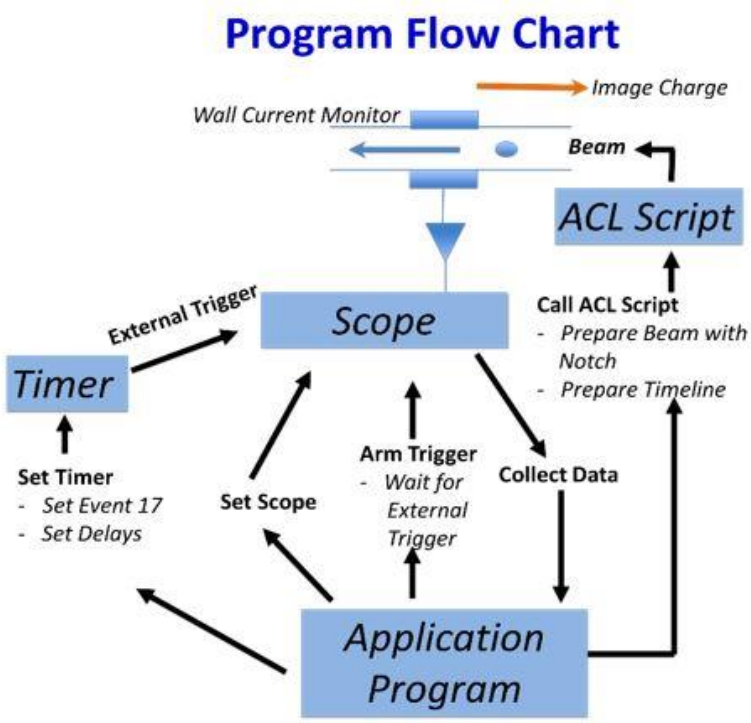

Figure 3: The Application program interacts with the Timer, which is responsible for sending an external trigger to the scope, the scope which reads from the Wall Current Monitor, and it calls an ACL Script that prepares the beam with a notch and sets a new timeline for the beam.

4) Once the requested beam event happens, it collects WCM data from the scope by triggering it w.r.t. $17.
5) Analyze the data to extract the beam energy spread.
6) Give a graphical output on a separate window for the user's perusal.

Some highlights of this programs functionality is, it is the first application program at Fermilab that can measure the beam energy spread of a multi- turn beam in the Booster at injection. In the past, the beam energy spread was measured only on a partial beam and only during a dedicated beam study. This program gives an ability to measure the beam energy spread with minimum interruption on the operation. Further, the instant measure of beam energy

spread at injection helps us to improve the overall Booster performance.

Analyzing the Data

The online analysis of the WCM data is one of the critical task of this application program. The program has to make several decision on the fly like, a) searching for the first notch and the subsequent appearance of the notch on each revolution, b) determine notch width (95%), c) average revolution period, etc.,

Before developing ACNET application program a computer package which works on Python platform is developed [8] to analyze scope data taken in binary format by using B62 [9]. We tested every stage of data ACNET analysis routines against the Python code.

Following the notch

The data received by the WCM for 100 Booster turns around injection looks like that shown in Fig. 4. The many notches that are present in this graph are actually the same notch but at a

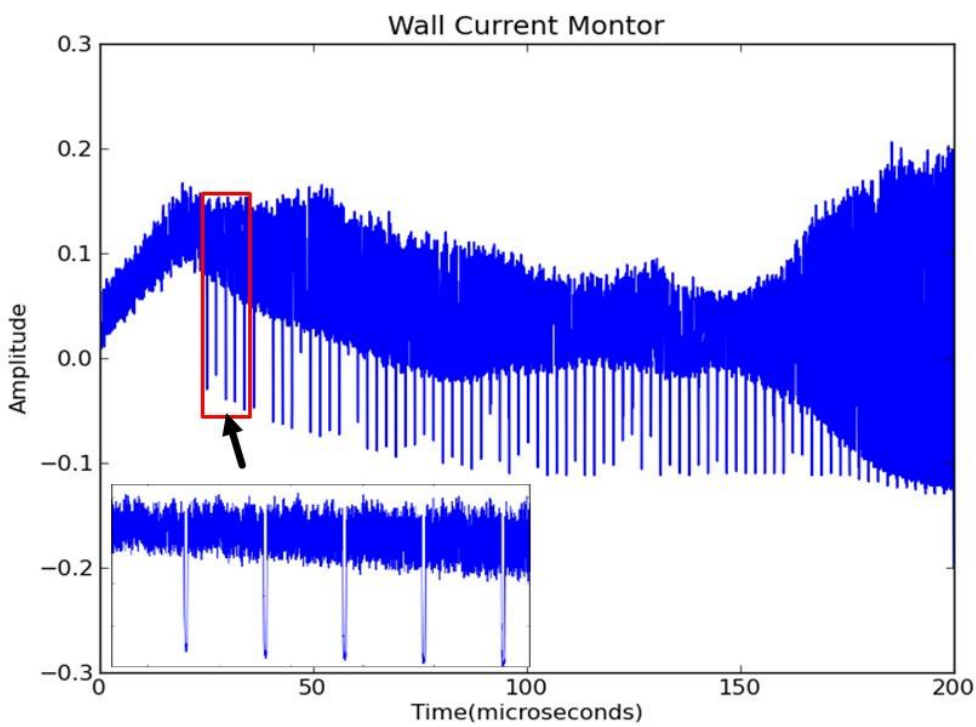

Figure 4: WCM data for the first 200 microsecond around injection. The icicles are the results of notch appearing at the WCM on every revolution. Zooming in it is possible to see a few of these notches, each is about 2.2 microseconds from each other (as shown in inset).

different revolution around the Booster. WCM is an AC coupled system. Because of a small amount of DC coupled part in it, the beam amplitude increases while it is being injected and then slowly starts to decline as it begins to settle down. Also after about 150 microseconds the RF in the Booster starts to ramp up; bunches are formed. The WCM data show an increase in signal amplitude as seen in Fig. 4 (around 200 micro second).

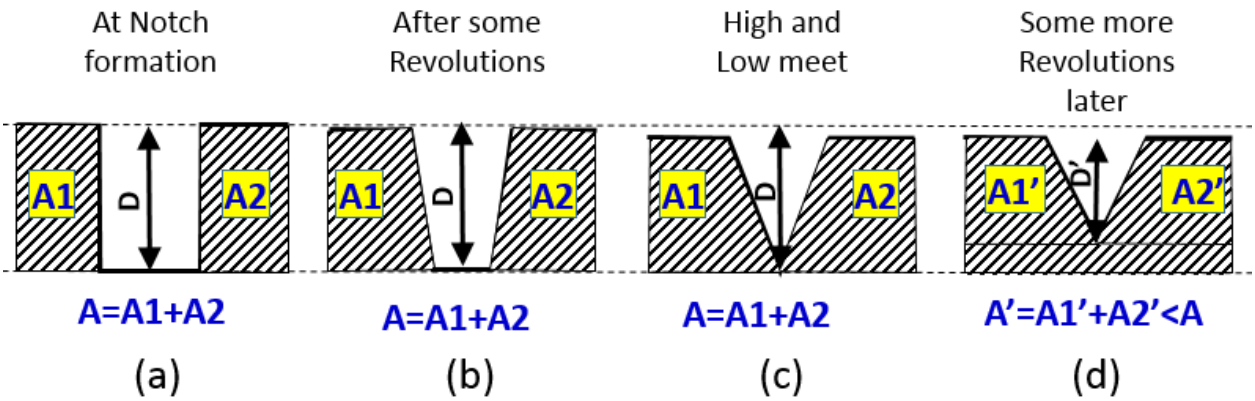

Figure 5: This is a schematic to explain the notch filling in an ideal system. a) at the instant of notch creation, b) the notch becomes less square and begins to 'fill up', c) the high and low momentum particles meet or 'touch', d) the integrated beam intensity appears to be decreasing after the baseline begins to increase right after they 'touch'. Notice that the areas remains the same till the high and low momentum particle meet (grazing touch). Also, the top level starts decreasing slowly as the notch gets filled up.

Over time this notch starts to evolve as seen by the WCM. As the particles drift in the notch the notch begins to 'fill up'. In Figure 5 we show a schematic (ideal case) of notch fill up. Until the grazing touch (for example see, picture "c" in Figure 5), the notch depth or integrated area between two successive notches in the WCM data remains constant. The depth of the notch start decreasing after the high momentum and low momentum particles "touch" (i.e., the level of the minimum start raising as shown in Fig. 5 (d)). Therefore, by following the minimum of the notch as the notch transforms, we are able to find the grazing touch.

A careful examination of WCM data reveals that the estimated area between two successive revolution of a notch as shown in Figure 5 is a better variable to search for grazing

touch than the depth of the notch.

Finding the First notch

First the program scans the entire WCM profile shown in Fig. 3 and then uses a guess period of 2.2114 microseconds to make "rough cuts". The program then searches for the minimum in each piece and saves those amplitude values and their corresponding time values.

Since we are interested in measuring the notch width accurately, first notch should be identified in the WCM data and studied in detail. To do this, the program compares all minima and looks for where a sudden step occurs. The time associated with first large excursion point represents occurrence of the first notch in the WCM data. The program infers that the subsequent minimum from the first notch are the notch at subsequent revolutions. Figure 6 depicts the "mountain range" plot of the WCM data around the notch. In this figure, the WCM trace in increment of 5 revolution are shown for clarity. In this data grazing touch has occurred at revolution number ≈30. Figure 7 shows rough cuts used to search for the minimum. Figure 8 shows results of such a search for the first notch in the data shown in Fig. 4.

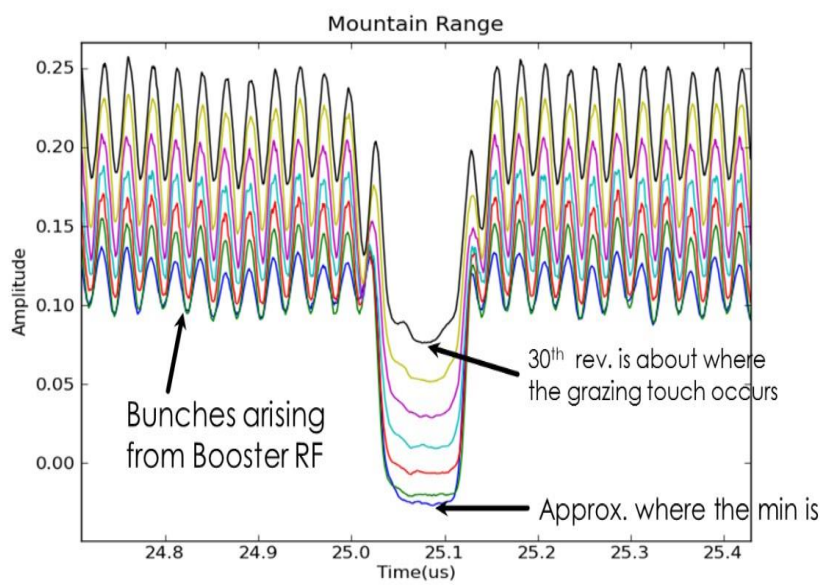

Figure 6: This is actual data from the WCM and is smoothen to show how the notch transforms over time. The blue notch is the first notch while the subsequent notches are every 5 revolutions until the 30th revolution. The presence of rf makes developing bunches and decision on grazing

touch difficult.

The downward trend in the plotted minimum amplitudes after 1st jump in Fig 8 (left) is arising from the similar feature seen in Fig.4; in an ideal case the minimum amplitude remain constant for some revolutions and start increasing. This feature makes the data analysis more difficult.

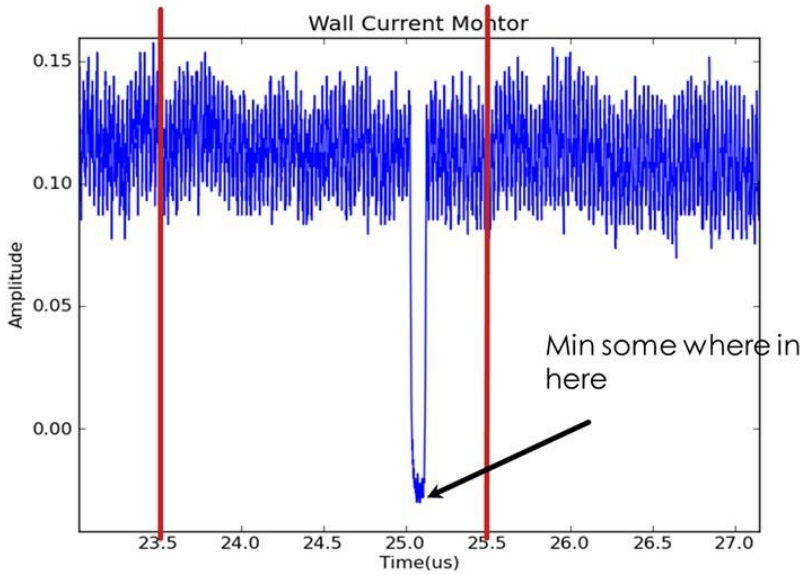

Figure 7: Rough cuts made by the program to find the minimum in segments of the data.

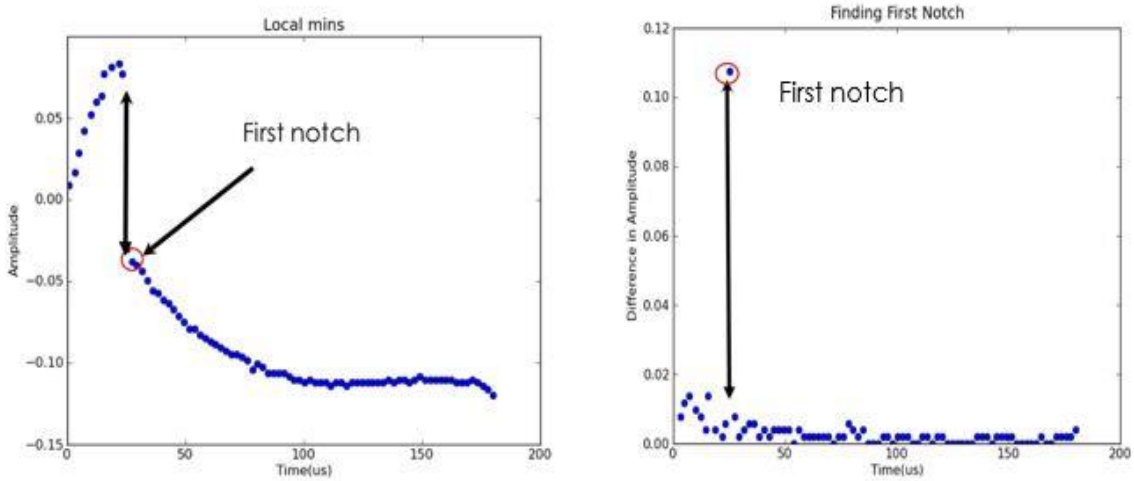

Figure 8: Left figure is a plot of all the minimum for every 2.2115 microseconds. The first notch appears after a large gap. In the right figure is the absolute difference of the next point and its 'left neighbor'. This is how the program identifies the first notch.

The revolution period is calculated using the first few notches. Figure 9 (left) shows revolution period versus number of revolutions. Using this calculated revolution period we can obtain better cuts of the data (see Figure 9 (right)). The large scatter observed in the measured revolution period as shown in Fig.9 (left), arises from the uncertainty in the location of the minimum in the notch region. Once we pass the grazing touch the location of the minimum is bunch better defined.

Next, by finding the area under the curve between two consecutive notches as shown in Figure 10 (left), we can plot the relative intensity versus revolutions (see Figure 10(right)). The base line for each revolution is calculated as an average of minimum amplitudes at notch. Data shows that, after the grazing touch the area sharply falls.

By eye one can easily determine where the curve in figure drops off. This inflection point is crucial because this is where the grazing touch occurs. After the grazing touch occurs the minimum amplitude increases (as shown schematically in Figure 5), therefore the area under the curve will decrease.

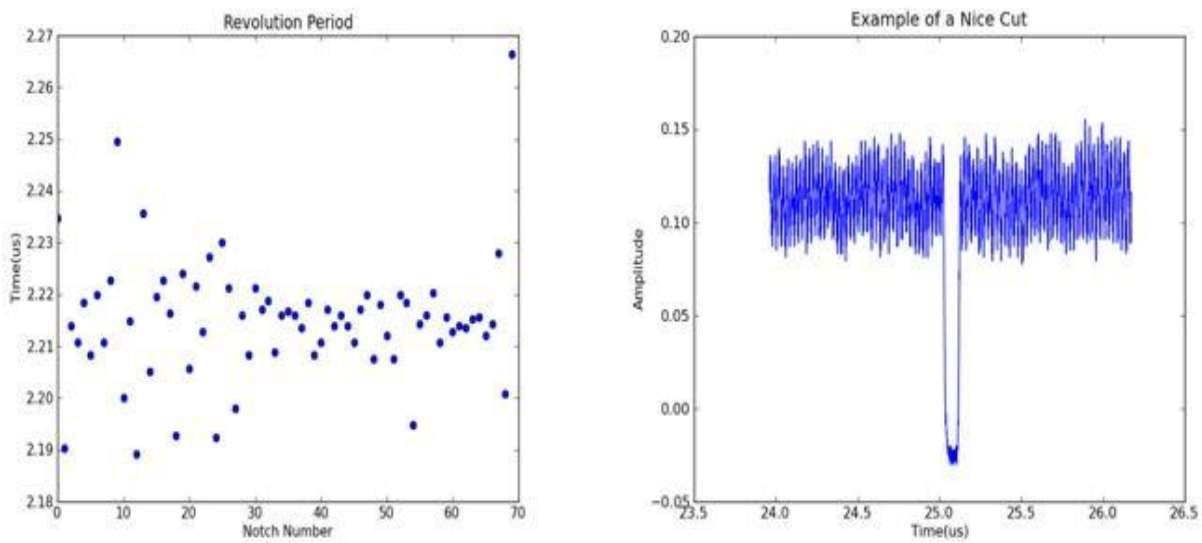

Figure 9: The plot on the left shows the revolution period of the notch as a function of turn number. They are a bit more scattered at the beginning because of noise. The plot on the right is an example of a nice cut made when a few of the revolution periods are averaged.

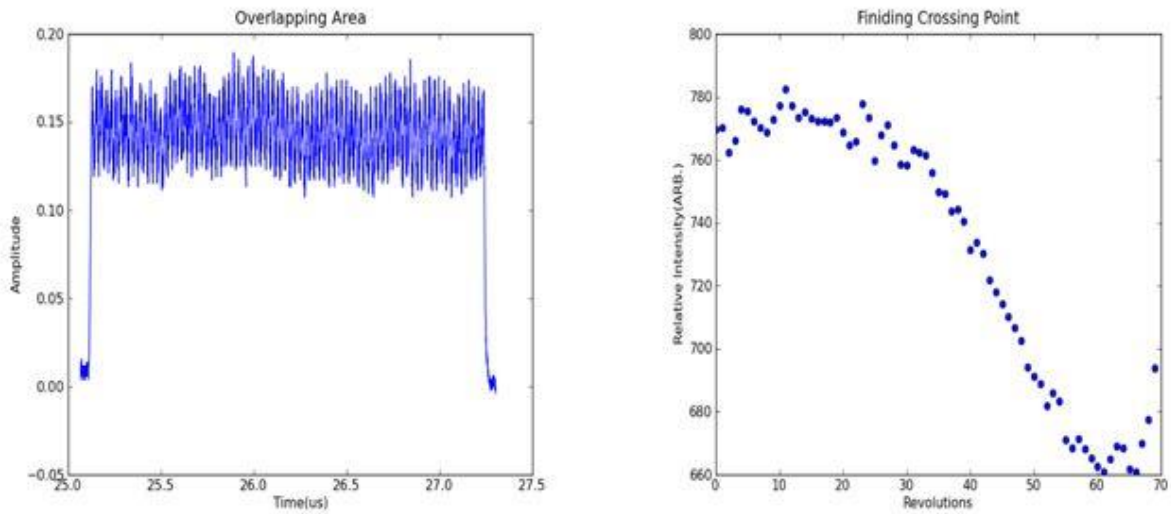

Figure 10: The plot on the left is slice between the first notch and the second notch. The area is taken under the curve as function of turn number is shown on the right. This plot shows how the notch transforms over time.

.

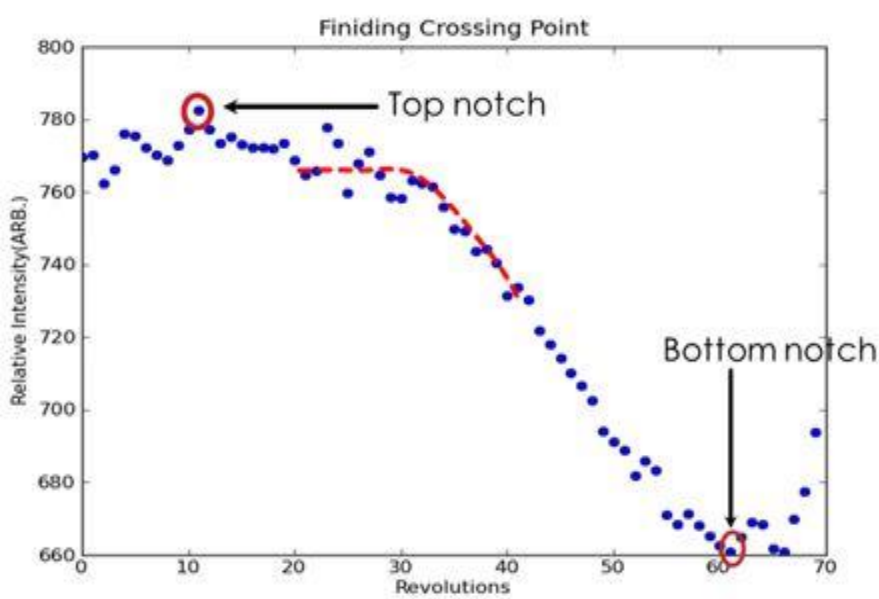

Figure 11: The red line is the region in which the program will most likely choose a grazing touch.

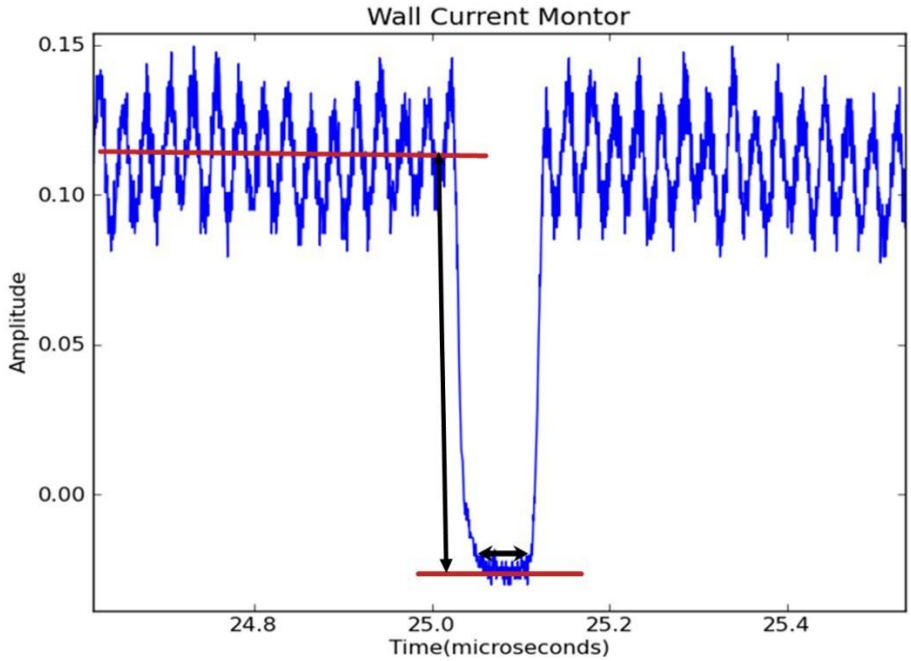

Figure 12: Width of the first notch

The program makes a decision based on an algorithm. It first calculates the approximate area of the Tgraze (which is the "Top notch"-("Top notch"-"Bottom notch")*.15) and then starting from the Bottom notch comparing each point to this approximate area of the Tgraze. When it reaches a value greater than or equal to this approximated value the program produces this value as the revolution number with a confidence level of about 1-2%. The "Top notch"(= maximum of calculated areas) and "Bottom notch" (= minimum of calculated areas) are indicated by circles in Figure 11.

Finding the notch width

In reality, even the first notch will not be a nice rectangular well. The shape depends on the kicker pulse shape which produce the notch. Consequently, measuring the notch width is one of the hardest task and introduces a large uncertainty in the measured energy spread. The notch width is found by taking an average of the baseline of the beam inside the notch where there is no beam and similar average of the beam level (see Figure 12). The difference between these two averages gives the depth of the notch. Then we determine the 95% width of the notch by

searching for 95% drop on both side of the notch. Such search is performed on the trace of the 1st notch as shown in Figure 12.

ACNET Page and Results

Figure 13 shows a typical front page of user interface upon opening. This comes with three windows: a) "Scope & Devices Config", b) "Plot Adjustment" and c) "Save and Recall files". In the "Scope & Devices Config." window various settings on the scope can be set. "set set up button will send the user set parameters to the scope. The "Arm Digitizer and Set Timeline"

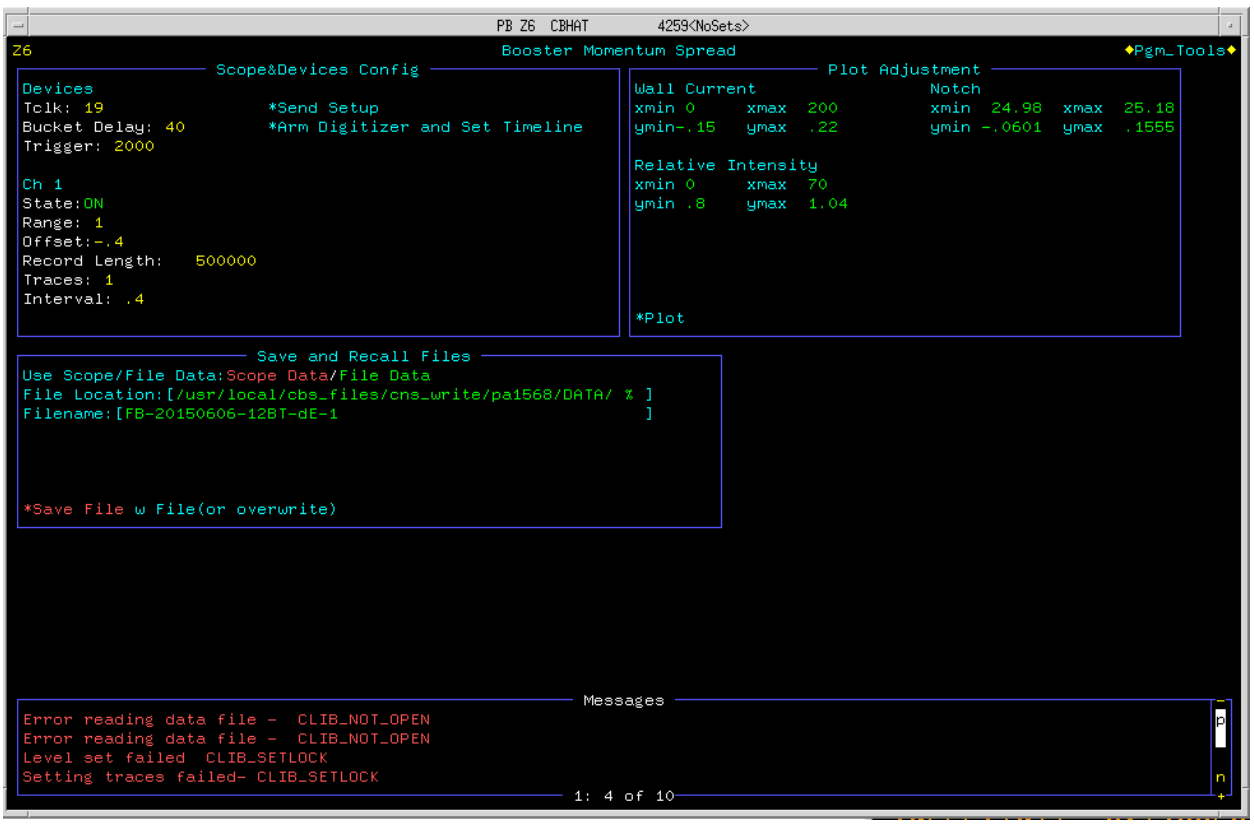

Figure 13: The user page of the Program. "Plot Adjustment" window is set to get Figure .14.

button arms the scope to wait for the external trigger and calls the ACL script. When this button is pushed a warning box appears that prompts the user to check with the MCR before proceeding. The program has the ability to plot data either straight from the scope or from saved files. One can adjust the plot sizes for the analysis in the "Plot Adjustment" window. The "Save

and Recall files" window gives the user to save new data if it is "Scope data" with a new name. On the other hand if it is "File data" (Old stored data) one can load it and perform offline analysis.

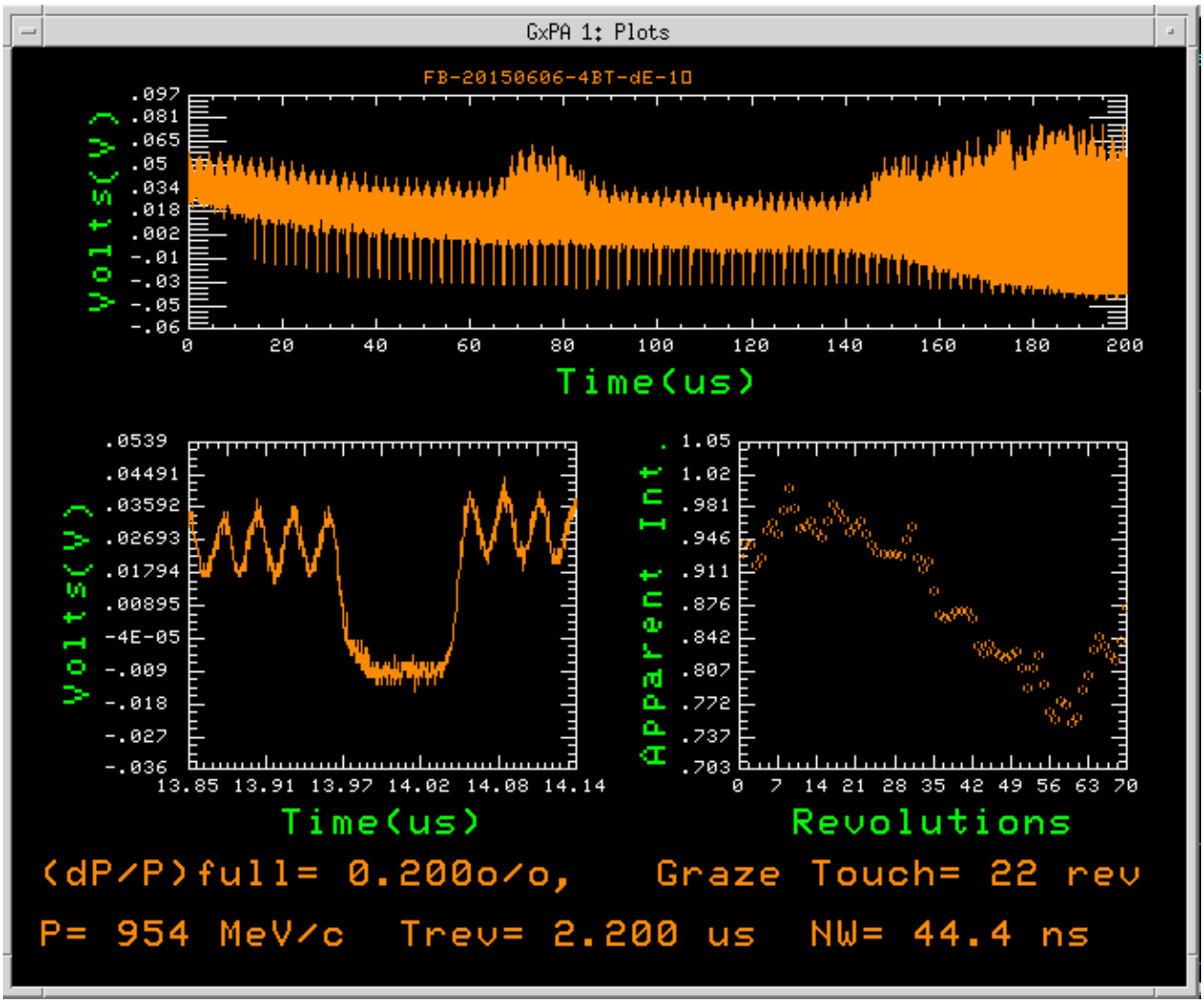

Figure 14a: The Analysis graphs. The top graph is the raw data, bottom left is the first notch, and bottom right is the apparent intensity graph normalized. The red box marks the dispalyed values useful for understading the energy spread. Data for FB-20150606-4BT-dE-1

Typical results of data analysis is shown in Figure 14a-14d. The raw data, 1st notch in the beam, and an apparent beam intensity used to determine the grazing touch are shown at top, lower left corner and lower right corner, respectively. At the bottom the values of the measured

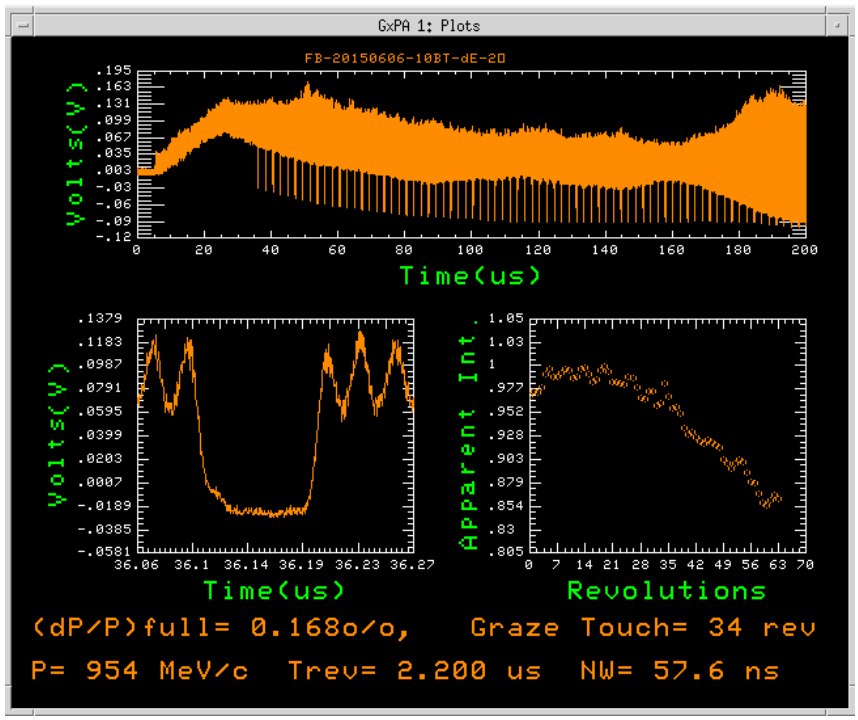

Figure 14b: Data for FB-20150606-8BT-dE-1. The Apparent area is less scattered.

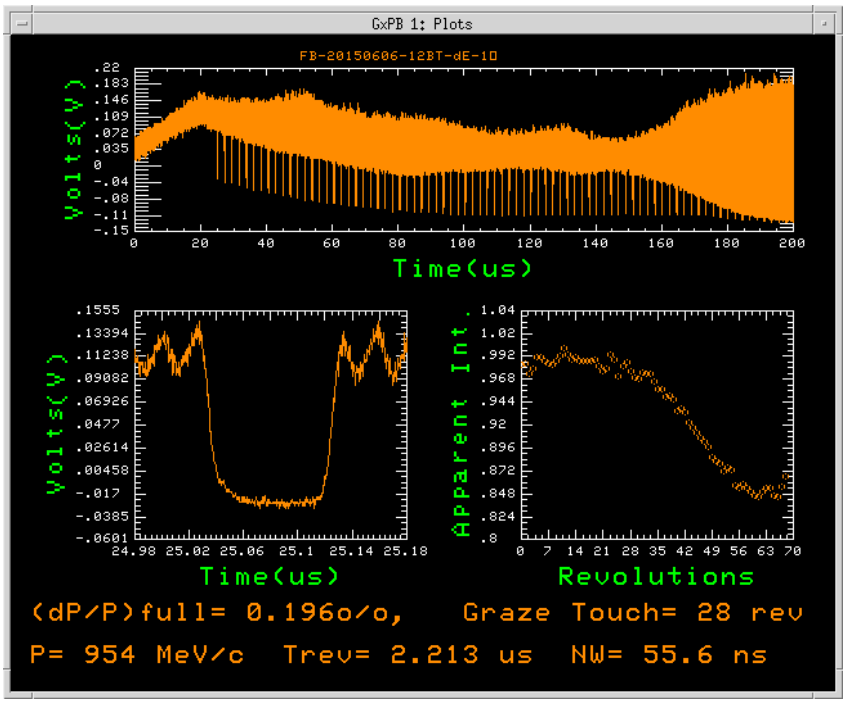

Figure 14c: Data for FB-20150606-12BT-dE-1. The Apparent area is less scattered.

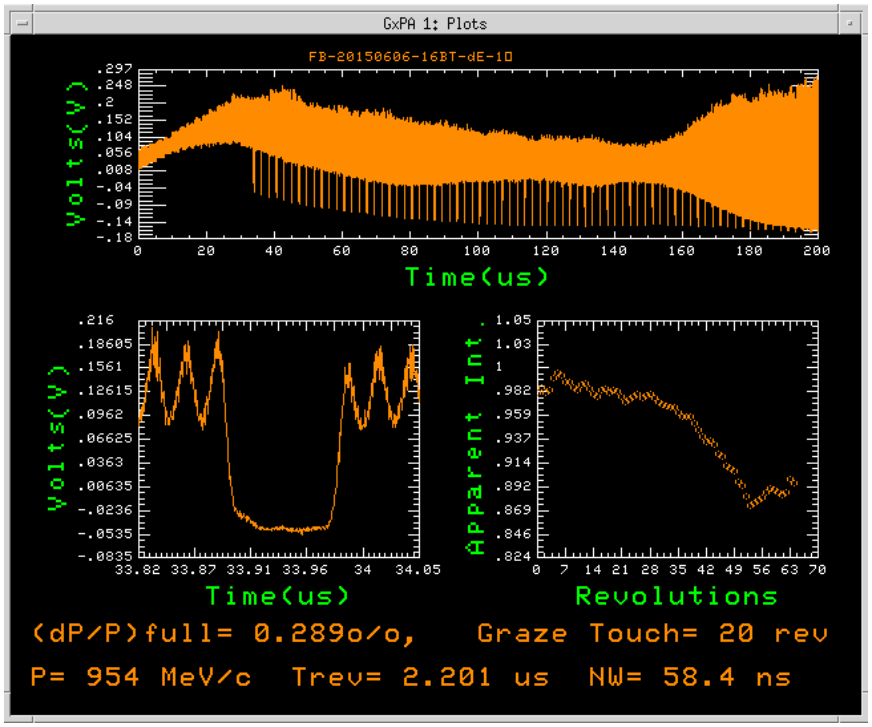

Figure 14d: Data for FB-20150606-16BT-dE-1. Though the Apparent Int. is less scattered the search routine found that the inflection point is at revolution 20 though by eye one finds this to be at around revolution number 30.

full momentum spread "dP/P", measured number of revolutions needed for grazing touch "Graze Touch", momentum of the synchronous particles "P", the measured revolution period "Trev", and the measured notch width "NW" are displayed. The middle two figures can also be used for a qualitative cross check for the final measurement values displayed in the bottom.

The data in Figure 14 clearly suggests that the routine developed here for notch width is quite robust. However, we need a better computer routine to search for the inflection point for grazing touch. Both these quantities depends on 1) proper choose of the range and offset settings in "Scope&Devices Cinfig," (reduced noise level) 2) stray rf field (smaller the better).

In Figure 15, the values of the energy spread are plotted for different Booster turns analyzed using the Python program where the inflection point for each case is determined case by case by visual examination and the current application program. We find quite good agreement between these two results. The data shows that we do not see a strong correlation

between beam intensity (each BT corresponds to about 0.37E12 protons per Booster cycle) and the beam energy spread at least up to 16 BT. The results from this is consistent with that presented earlier [6, 7].

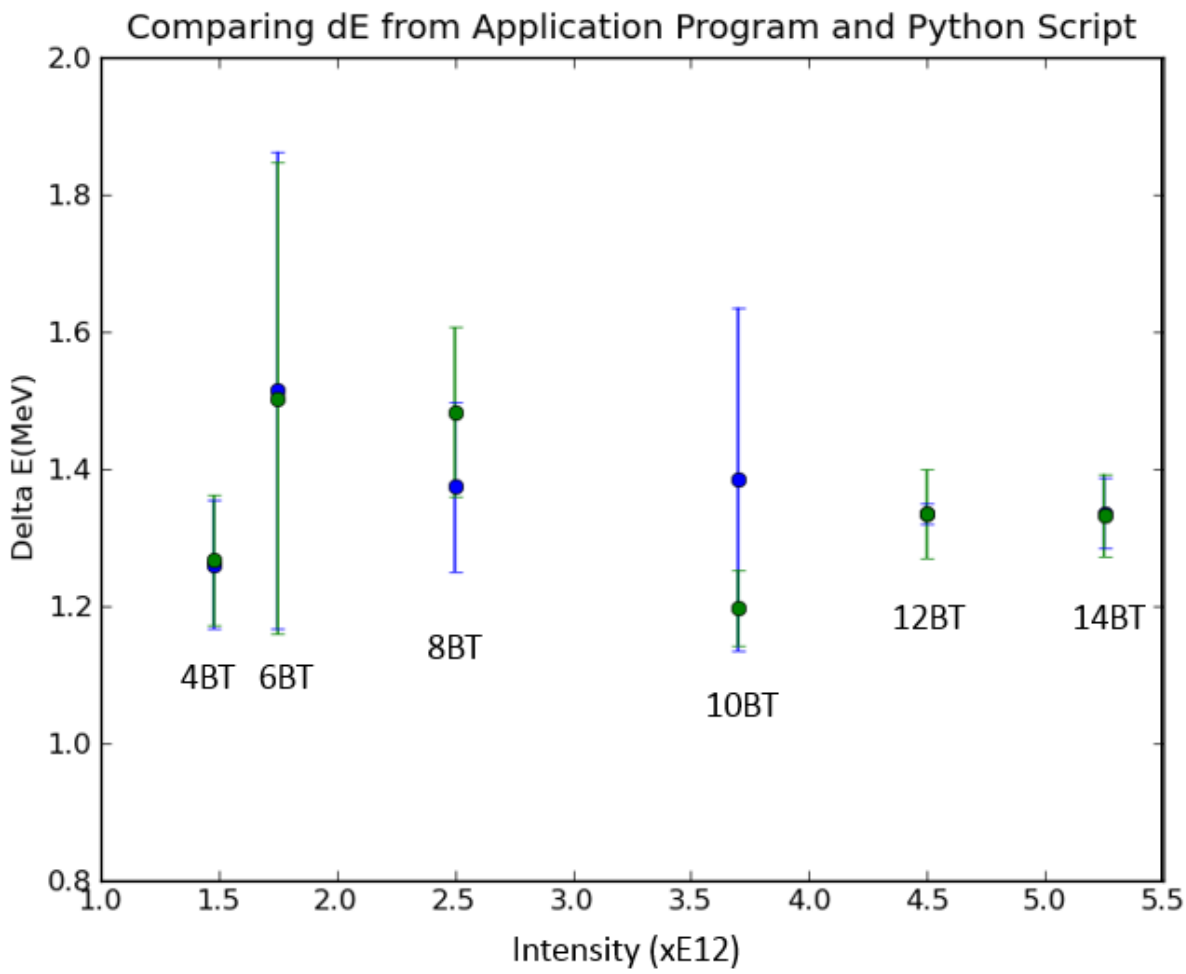

Figure 15: Measured full beam energy spread in the Booster at injection for different beam intensities. BT≈0.37E12 protons per Booster cycle. The error bars are maximum deviation from the mean for most of the data points except for 16BT (a 20% error is assumed).

Final Remarks

Understanding the energy spread of beam upon injection into the Booster is extremely useful in improving the beam performance in the Booster. This allows for quick online measurement of energy spread on multi-turn Booster beam thus giving a fuller picture of the beam. It also can then be used as comparison of how the energy spread from injection changes at

other points in the acceleration cycle. Therefore having an application program that can help identify gradual drifts and/or sudden change in the upstream LINAC and/or transfer line elements that can cause changes in the energy spread is very useful in improving and maintaining the quality of the beam. Further, the shape of the notch profile as seen by the beam can also be used as another diagnostic tool.

Acknowledgement

We would like to thank K. A. Triplett for his help in beam studies and Dr. R. Ainsworth for his help in creating a Python routine to convert scope data in binary format to CSV format.

References

* SIST Summer intern from Brown University.

[1] R. R. Wilson, Proc. of the 8th Int. Conf. on High Energy Accel. 1971, Geneva, Switzerland, p3; R. Billinge et. al, IEEE Trans. Nuc. Sci. 1969 Particle Acce1. Conf. p. 969.
[2] Booster Rookie Book, http://www-bdnew.fnal.gov/ operations/rookie_books/ rbooks.html.
[3] W. Pellico, et al., IPAC2014 (2014) p 3409; R. Webber et al., Beams Document 3781-v2 (2011).
[4] "The PIP-II Reference Design Report," V1 March 2015, http://www-bdnew.fnal.gov/pxie/PIP-II_RDR/ PIP-II_RDR.pdf.
[5] M. Popovic, (private communications, 2014).
[6] C. M. Bhat, et al., IPAC2I015 (2015), THPF113.
[7] C. M. Bhat (2015, unpublished).
[8] J.J. Nelson and C. M. Bhat (Python package 2015, unpublished).
[9] ACNET Application Program B62, Ming-Jen Yang (private communications, 2014).