Bunch By Bunch Beam Instrumentation at CESR

Vishwajit.Y.Vyas

Wayne State University, Detroit, MI, 48201

(Dated: August 13, 2005)

The Cornell Electron Storage Ring (CESR) is equipped with a new beam Position monitor (BPM) readout system that provides local bunch-by-bunch and turn-by-turn processing of the data. The system is based on processing the data gathered from the BPM’s on its on board DSP. This project deals with enabling the calculation of the FFT of the data that is gathered for each bunch, for upto 1024 turns. The analysis of the resultant frequency spectrum is helpful in determining the beam parameters like its horizontal and vertical tunes. These are the betatron frequencies for horizontal and vertical motion.

I. INTRODUCTION

The beam at CESR follows a circular path of motion due to the presence of the bending magnets and it is kept focused by the focusing magnets. So it does not have many degrees of freedom as far as the motion is concerned. However, the beam undergoes some oscillations in the horizontal and vertical planes. The frequencies at which the oscillations occur are characteristic of the focusing optics used in the ring. The capacitive pickups in the beam buttons inside the BPM measure the position of the beam in the beam pipe. Thus, it becomes possible to measure these different frequencies or "tunes". The BPM system measures the trajectory (closed orbit) of the beam around the ring and can be used to measure transverse oscillations around the closed orbit. The DSP’s on board the BPM systems enable us to make these measurements in real time for individual bunches in the ring and thus are very useful.

II. DEVELOPMENT

The entire process of performing an FFT and observing the resultant frequency spectrum was performed on a spreadsheet\(^1\). This method was tested on the old data taken from the ring. Once the tests proved that the results from the spreadsheet agreed with the observed values, the process of creating a C program to calculate the FFT values was started. In addition to doing so, the program also finds the peaks and the average of the values. The output of this C program was again matched up against the ones generated by the spreadsheet. Additional testing was done to ensure that the entire code was "bug-free". The figure1 and figure2 display the frequency spectrum for the X and Y values, with the Amplitude in dB. Once the C program gave satisfactory results, the next step of combining this new code into the existing DSP modules began. This was by far the most challenging aspect of the whole project. Several changes needed to be made to free up the required code memory and data memory. Also, the C code had to be modified so that it was modular in nature and could be merged easily with the existing DSP codes.

\(^1\) created by Gerald Codner, Cornell University
FIG. 1: Plot for the Frequency Spectrum for X values.

FIG. 2: Plot for the Frequency Spectrum for Y values.

III. PROCEDURE

1. The X and Y values are calculated from the beam button values. These position values are then used throughout the process for finding the FFT values. The number of such points has to be n, where n is a power of 2. If $S_0, S_1, S_2$ and $S_3$ are the button signals, then the corresponding x and y position values are calculated as follows: In the linear approximation, the transverse position can be written as:

   \[
   \Delta x = [S_0 - S_1 - S_2 + S_3] \\
   \Delta y = [S_0 + S_1 - S_2 - S_3] \\
   x = c \frac{\Delta x}{\Sigma} \\
   y = c \frac{\Delta y}{\Sigma}
   \]

   where $c$ is a constant determined from the geometry of the beam.

2. Next, the windowing function values are calculated for the specified number of data points. Presently, a Gaussian is used as a windowing function. The weight function $W(t)$ looks as follow:
\[ W(t) = e^{-\left[\frac{t-n/2}{n/4}\right]^2} \]

The use of windowing function smoothes out the data so that the peaks are well defined in the spectrum. The following figure shows the Gaussian for 1024 points.

3. Each position value is multiplied by its corresponding value of the windowing function and the entire range is passed to a function that calculates the fourier transform of the data.

4. The output of the function is in the form of complex numbers. Also, the total number of FFT points in the output is reduced to \( n/2 \). Their magnitudes are calculated and a scale factor is applied. This scale factor is essential to restore the correct amplitude information in the output.

5. Now this set of numbers is passed to a function that finds out the specified number of peaks in the given range, and their indices. The function neglects a certain number of beginning and ending values, as specified. The total range of the spectrum is 390 KHz, which is reduced to half its value by the FFT, and is spread over \( n/2 \) index points. So the actual frequency of the peaks can be found by applying the proper calculations. A maximum of ten peaks and indices can be found for each plane.

IV. SIMULATION

The testing of the project was divided into two stages. In the first stage, the code was tested on a DSP simulator. Several changes had to be made to the code in order to make it work as desired on the DSP. This testing ensured the robustness of the code and increased its reliability. After sufficient testing on the simulator, the code was ready to be tested on the actual DSP’s in the ring. To facilitate this, several additions were made to the code. The communication protocol and tracing features were added. The input and output structures were modified to comply with the existing conventions. The various status and error flags were inserted.
V. OPTIMIZATION

Once the algorithm was working satisfactorily, several changes were made to the code so that the processing could be as fast as possible. The functions used were made as streamlined and robust as possible. The function calls were minimized so that the overheads associated with them were limited. The various parameters to be passed into the functions were scrutinized and modified if seemed fit. Also, the conversion of the amplitude values into dB units had to be removed from the code, due to the processing time associated with it. All these optimizations enabled the fast execution of the code, while using as little memory and processor cycles as possible.

VI. FURTHER TESTING

The code was again passed through various other simulation tests to check for the effects and potential bugs introduced due to optimization. The communications aspects of the code were verified and tested. The control system version of the code was built on the VMS platform, to enable cross-checking of the results, and was tested also.

VII. HARDWARE TESTING

Now the code was finally downloaded into the DSP, to enable real time hardware testing on it. This process was started on 11th August, 2005, when the first results of the tests were obtained. Still further testing needs to be done for verification of the results.

VIII. RESULTS

- The simulations and the hardware tests prove that the algorithm works according to the specified needs. The plots of the output generated by the program in the first hardware tests are shown below.
- The Timing Test Results:
  The timing tests measure the time required by the program to get raw data, process it, and save the output, for a 1024 turn sample. The optimal frequency at which this can happen is 80.9Hz. This is also the proof that the optimization of the code was useful in reducing the processing time.

IX. ACKNOWLEDGMENTS

- I would like to thank the NSF-PHY0353994 grant for the funding.
- It is my pleasure to acknowledge Mark Palmer, of Cornell University, for providing me with an engaging and exciting project, for enduring my constant interruptions, and guiding me through along the way. I wish to thank Gerald
FIG. 3: Plot for the results in X plane.

FIG. 4: Plot for the results in Y plane.

Codner for helping with the spreadsheet analysis and explaining to me the basics of FFT and spectrum analysis. I also thank Charlie Strohman and Eugene Tanke for their help and support. I also wish to thank R. Galik, Cornell University, and G. Bonvicini, Wayne State University, for organizing what is, by far, the best summer experience I have had.