DSP Algorithm for the Closed Orbit Analysis of the Signal from the Accumulator BPMs

G. Garzoglio, K. E. Gollwitzer and G. Stancari

August 19, 1999

P-Bar Note #615

1 Introduction

A new algorithm has been developed for the digital signal processor (DSP) in the digitizing card of the Accumulator beam position monitor (BPM) system. The main motivation is to make the measurement more accurate. The previous algorithm consisted of a fast Fourier transform (FFT) with a peak search; this was implemented only using 8192 samples. The algorithm presented here uses the ACNET value of the bunching RF frequency as an input parameter and so it only needs to perform a one-point Fourier transform. Moreover, a ping-pong mechanism for the management of the DSP memory has been implemented, which allows exploitation of a larger set of digitized data.

2 The BPM Readout System

The signal from the BPM pick-up plates goes through a pre-amplifier mounted on the beam pipe at the location of the pick-ups. The pre-amplifiers also have a calibration input which can be used to determine a normalization between channels used by the DSP algorithm. After being brought up to the service building, the signal enters an analog down-converter (DC) card, which consists of an amplification stage and two low-pass filters (figure 1) for closed orbit measurements. Note that there is no down conversion done when the BPM system is in Closed Orbit Mode. The output of the DC card is fed to the digitizer card, which includes a sampling ADC and a DSP. A console application passes parameters to the DSP and reads the resulting position and amplitude measurements through the Slot 0 VXI Controller.

For every sector there is a VXI crate with four sets of analog and digitizing cards. Each set of cards consists of eight channels allowing for the determination of the position and amplitude for four BPMs. Each DSP is responsible for calculating the position for four BPMs. There are a total of 24 DSPs in the BPM system.

More information can be found at `http://www-rfi.fnal.gov/PBBPM/PBBPM.html`. In particular, information about ACNET control is documented.

3 Measurement Technique

The quantity of interest is the amplitude of the signal from each of the pick-up plates of a BPM. Let us first assume that, due to the RF bunching, the signal as a function of time is described by the function $s(t) = A \sin(2\pi f_{\mathrm{RF}} t + \phi)$. The first step of the algorithm (which we call "heterodyne method" in analogy with the super-heterodyne radio receiver) is to modulate the signal with a sine and a cosine of known frequency f_m (chosen to be as close as possible to f_{RF}):

$$s_{\sin}(t) \equiv s(t) \cdot \sin(2\pi f_m t)$$

$$s_{\cos}(t) \equiv s(t) \cdot \cos(2\pi f_m t).$$

The resulting functions $s_{\sin}$ and $s_{\cos}$ are the sum of a low-frequency ($|f_{\mathrm{RF}} - f_m|$) and a high-frequency ($f_{\mathrm{RF}} + f_m$) component. They are then integrated over a number n of periods of the modulating frequency f_m:

$$A_{\sin} \equiv \frac{2}{(n/f_m)} \int_0^{n/f_m} s_{\sin}(t)dt$$

$$A_{\cos} \equiv \frac{2}{(n/f_m)} \int_0^{n/f_m} s_{\cos}(t)dt.$$

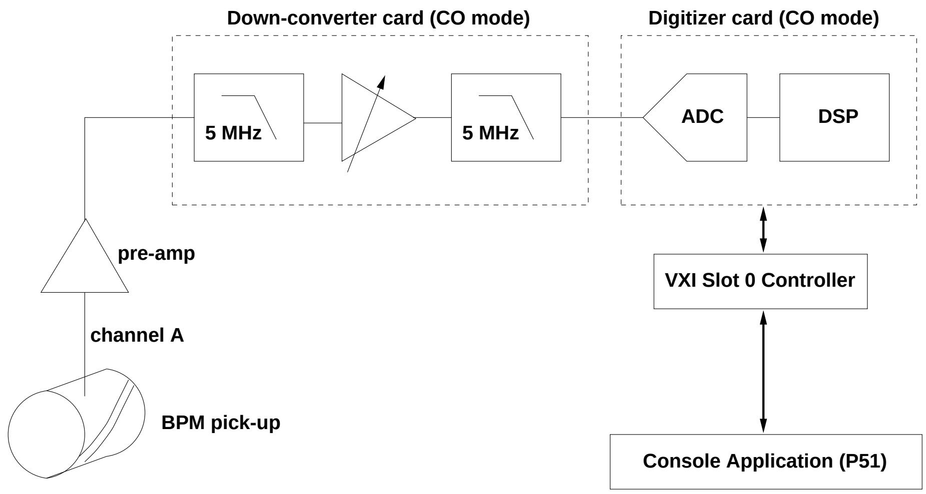

Figure 1: BPM system readout schematics (closed-orbit mode).

The number n is chosen according to the following criteria:

$$1 \ll n \ll \frac{f_m}{|f_{\rm RF} - f_m|};$$

in this way, the integration removes the high-frequency component of $s_{\rm sin}$ and $s_{\rm cos}$. The geometrical average of the two integrals $A_{\rm sin}$ and $A_{\rm cos}$ represents the measurement of the signal amplitude A:

$$A_{\rm meas} \equiv \sqrt{A_{\rm sin}^2 + A_{\rm cos}^2}\ . \tag{1}$$

Multiplying the signal by both sine and cosine is done to get rid of the initial phase ϕ. This procedure is equivalent to calculating the modulus of the Fourier transform of the function

$$s(t) \times (\text{square box of length } n/f_m)$$

at $f = f_m$:

$$A_{\rm meas} \propto |\mathcal{S}(f_m)| = \left| \int_0^{n/f_m} s(t)\, e^{i2\pi f_m t}\, dt \right| .$$

If the modulating frequency f_m is not equal to $f_{\rm RF}$, the measured amplitude $A_{\rm meas}$ is smaller than the actual amplitude A. Depending upon the desired precision on the measurement of the amplitude, one can estimate a tolerable value for the frequency difference $f_m - f_{\rm RF}$, what precision $f_{\rm RF}$ must be known. In the case of sinusoidal signal, the ratio $A_{\rm meas}/A$ is given by the following expression (plotted in figure 4(a) as a function of $f_m - f_{\rm RF}$ for $f_{\rm RF} = 1.27$ MHz, $n = 3251$ and $\phi = 0$ rad):

$$\begin{aligned} \frac{A_{\rm meas}}{A} &= \frac{f_{\rm RF}}{f_m} \cdot \frac{2f_m}{f_{\rm RF} + f_m} \cdot \left| \frac{\sin\left[n\pi\left(1 - \frac{f_{\rm RF}}{f_m}\right)\right]}{n\pi\left(1 - \frac{f_{\rm RF}}{f_m}\right)} \right| \\ &\times \sqrt{\cos^2\left(\phi + n\pi\frac{f_{\rm RF}}{f_m}\right) + \frac{f_{\rm RF}^2}{f_m^2}\sin^2\left(\phi + n\pi\frac{f_{\rm RF}}{f_m}\right)}\ . \end{aligned} \tag{2}$$

Besides its obvious maximum at $f_m = f_{\rm RF}$, this function has a series of zeroes, the closest of which occur at $|f_m - f_{\rm RF}| = f_m/n$. This is just a quantitative way of saying that the wider the integration interval is, the more sensitive to a frequency error the amplitude measurement will be. Even when $f_m = f_{\rm RF}$, there is an error if the integration extends over a non-integer number of periods $\nu \equiv n + \varepsilon$ $(0 < \varepsilon < 1)$, but it is usually much smaller and decreases rapidly as n increases. In the case of a sinusoidal signal, it can be evaluated as follows:

$$\frac{A_{\rm meas}}{A} = \left(1 + \frac{\varepsilon}{n}\right)\sqrt{1 + \frac{\sin 2\phi - \sin\left(2\phi + 4\pi\nu\right)}{2\pi\nu} + \frac{1 - \cos 4\pi\nu}{8\pi^2\nu^2}}\ .$$

To estimate how good the measurement of the amplitudes A and B on the two pick-up plates has to be in order to get a reasonable position accuracy, the relationship between the position x and the ratio $r = B/A$ has been analyzed[1]:

$$x = \frac{A-B}{A+B} = \frac{1-r}{1+r}.$$

Figure 2(a) emphasizes the dramatic change in amplitude ratio as the position spans the available range. An amplitude measurement error is parameterized by a factor α (which will be, in general, a function of position) multiplying the ratio r: $\alpha = r_{\rm meas}/r$. Figure 2(b) shows the position error Δx as a function of r for different values of α. As can be expected, the most stringent constraints on accuracy come from positions near the center of the BPM. From the same figure, one can deduce that if the ratio of the amplitudes is measured with 5% accuracy, the position error is always smaller than about 2% of the full effective radius.

In reality, the BPM signal, instead of being a smooth function $s(t)$, consists of a string of $2^{17} = 131072$ integer numbers, s_i, in the range 0–4095, which come from the 12-bit sampling ADC and are stored in the memory of the digitizer card. The sampling frequency has been chosen to be $f_s = 25.6$ MHz, so that each oscillation period of the signal contains $f_s/f_m \simeq 20$ sampling points (the RF is $f_{\rm RF} = 1.27$ MHz, twice the revolution frequency of the beam). Therefore, each data set has information on approximately $6 \cdot 10^3$ oscillation periods. In the DSP code, the measurement of the signal amplitude is achieved through the

[1]Neglecting the effective radius and the position offset of the pick-up, or the relative calibration between the two plates, does not affect the generality of the following remarks.

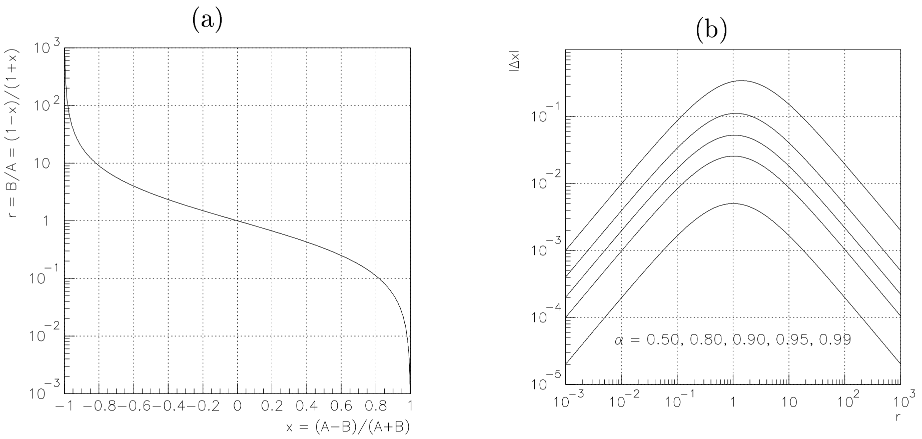

Figure 2: (a) The ratio B/A as a function of position x. (b) Position error as a function of the ratio B/A for five different values of the error α on the ratio: top curve $\alpha = 0.5$ and bottom curve $\alpha = 0.99$.

following formula, in analogy with equation. 1:

$$A_{\text{meas}} = \frac{2}{N_S}\left\{\left[\sum_{i=1}^{N_S} s_i \cdot \sin\left(2\pi\frac{f_m}{f_s}i\right)\right]^2 + \left[\sum_{i=1}^{N_S} s_i \cdot \cos\left(2\pi\frac{f_m}{f_s}i\right)\right]^2\right\}^{1/2}.$$

The number N_S is chosen in order to include the desired number of complete oscillations; in our simulations and measurements it usually varies between 8192 and 65536 (a memory hardware problem prevented us to use all of the 131072 points). To save computing time, sines and cosines are calculated only once and stored in a lookup table in the DSP memory. The lookup table has 15000 entries, which are sufficient for representing one period of a sinusoid with good numerical accuracy.

An averaging feature has also been implemented ("video averaging"). For a given total number N_T of sampling points (which is usually the maximum number of available data points 65536), one can specify N_S ($1 \le N_S \le N_T$) and the offset between samples N_H ($1 \le N_H \le N_S$); their meaning is explained in figure 3. In this case, more than one amplitude measurement is performed, one for each of the samples in figure 3; the final result is the arithmetic mean of these measurements. The purpose of the video averaging feature is to reduce the spread of the position determination while keeping it independent from the difference $f_m - f_{\text{RF}}$ (within a desired frequency range).

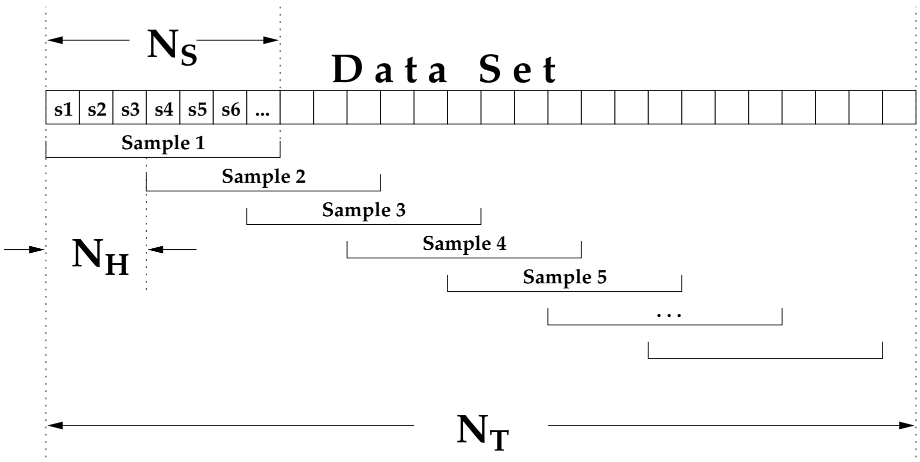

Figure 3: Diagram showing the averaging procedure.

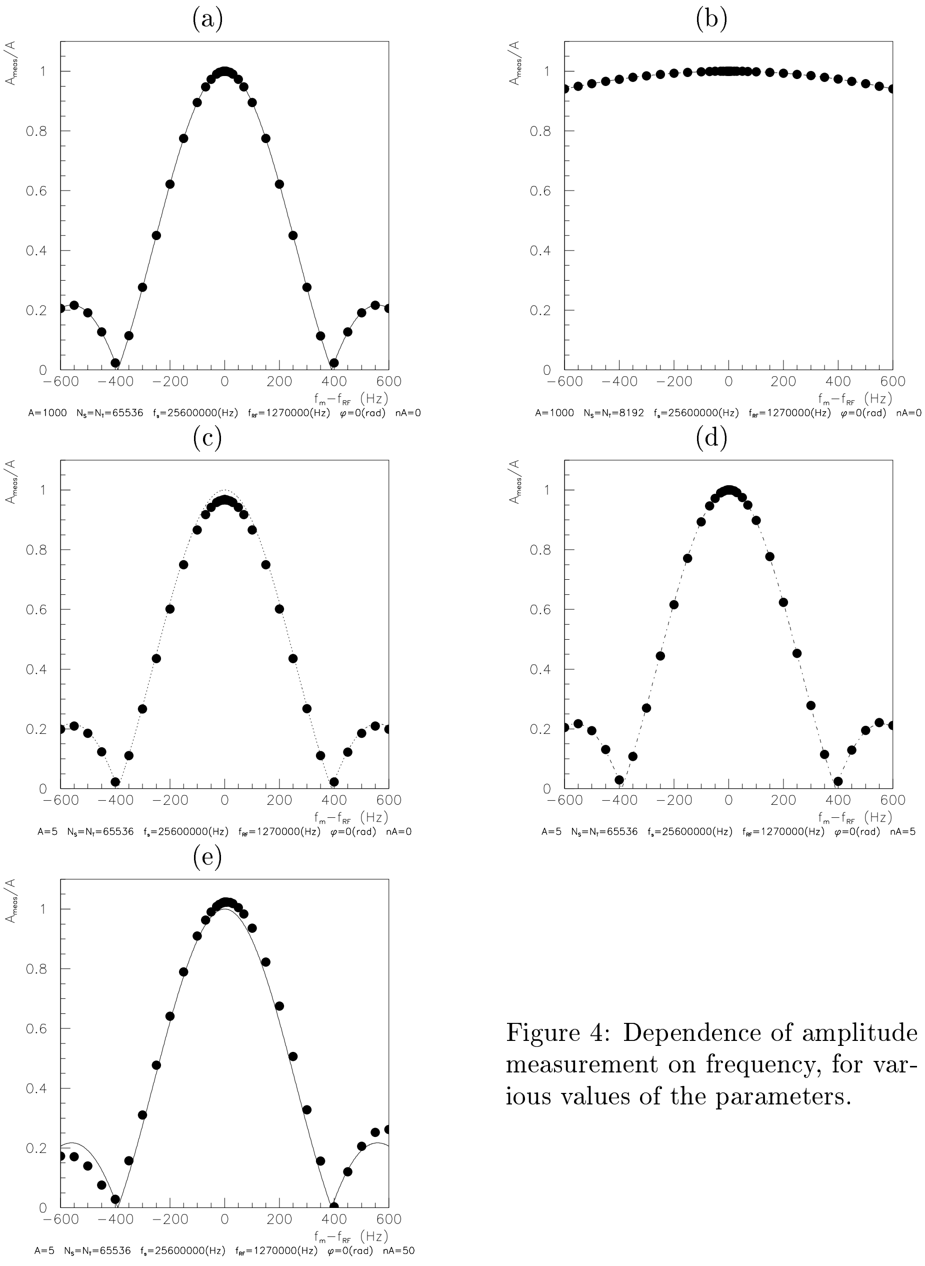

Figure 4: Dependence of amplitude measurement on frequency, for various values of the parameters.

4 Simulations

The algorithm has been tested first by giving it a fake input data sample, generated by a software signal simulator. The simulator adds white noise[2] to a sinusoid of given amplitude, and then discretizes the result, producing an ADC-like output (a set of integer numbers). Besides the signal and noise amplitudes, all the other parameters (sampling and modulating frequency, number of data points, ...) can be changed to see their effect.

Figure 4 shows some of the results of the amplitude measurements for the simulated signal. The solid dots represent the output of the algorithm, while the curves are the expected behavior as a function of f_m (equation 2). The signal amplitude A and the noise amplitude n_A are in ADC counts. In the best case, when the amplitude is large compared to the LSB of the ADC and there is no noise, the predicted amplitude is identical to the measured amplitude, as shown in figure 4(a). Figure 4(b) confirms that integrating over a smaller number of periods, the measured amplitude becomes less sensitive to the frequency difference. In figure 4(c) is shown the effect of discretization in the ADC; the amplitude of the signal has been chosen to be only 5 ADC counts. An amount of random noise comparable to the signal can compensate the effects of discretization, as shown in figure 4(d). Figure 4(e) illustrates that larger noise will distort the results.

The position measurement has also been simulated. Four positions have been chosen (0 mm, 5 mm, 20 mm and 50 mm, with an effective radius of 68 mm), and four noise amplitudes $(0, 1, 5$ and 20 ADC counts). The software signal simulator generates a pair of signals that would give the desired position; this is done at nine different values of the amplitude of the largest channel (channel A), from about 1 ADC count to about 100 ADC counts (represented by the simulated gain of channel A going from $g_A = -10$ dBV to $g_A = 30$ dBV). The pair of signals is the input of the heterodyne algorithm, which outputs a measured position. The measured position is then compared with the position generated by the simulator. For each value of the above-mentioned parameters, twelve different pairs of signals are generated, so that the spread of the results can be evaluated. Figure 5 shows the measured position as a function of amplitude for the four different values of the noise amplitude; for these plots, $N_T = 65536$ and $N_S = N_H = 8192$. The spread in the measurements is quantitatively represented by the standard deviations in figure 6. This simulation predicts that, if the white noise is small (say, 1 ADC count) and the only other source of non-linearity is the ADC digitization, the position can be determined to less than 0.3 mm (using all the available data points $N_T = 65536$).

Note that none of the simulations have any coherent noise. The actual environment has many sources of electronic noise.

[2]A random number is generated, with Gaussian distribution. The mean of the distribution is zero, whereas its standard deviation is equal to the adjustable parameter n_A.

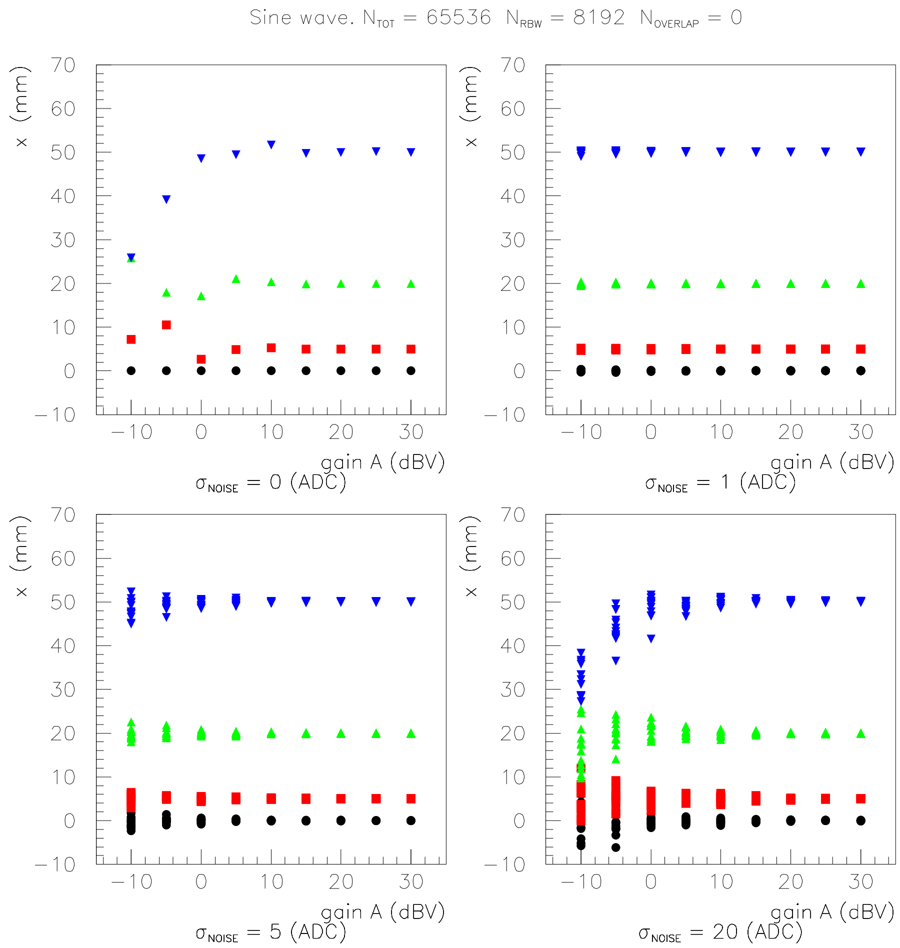

Figure 5: Results of the simulated position measurements. The generated position was 0 mm (black dots), 5 mm (red squares), 20 mm (green triangles pointing upwards) and 50 mm (blue triangles pointing downwards).

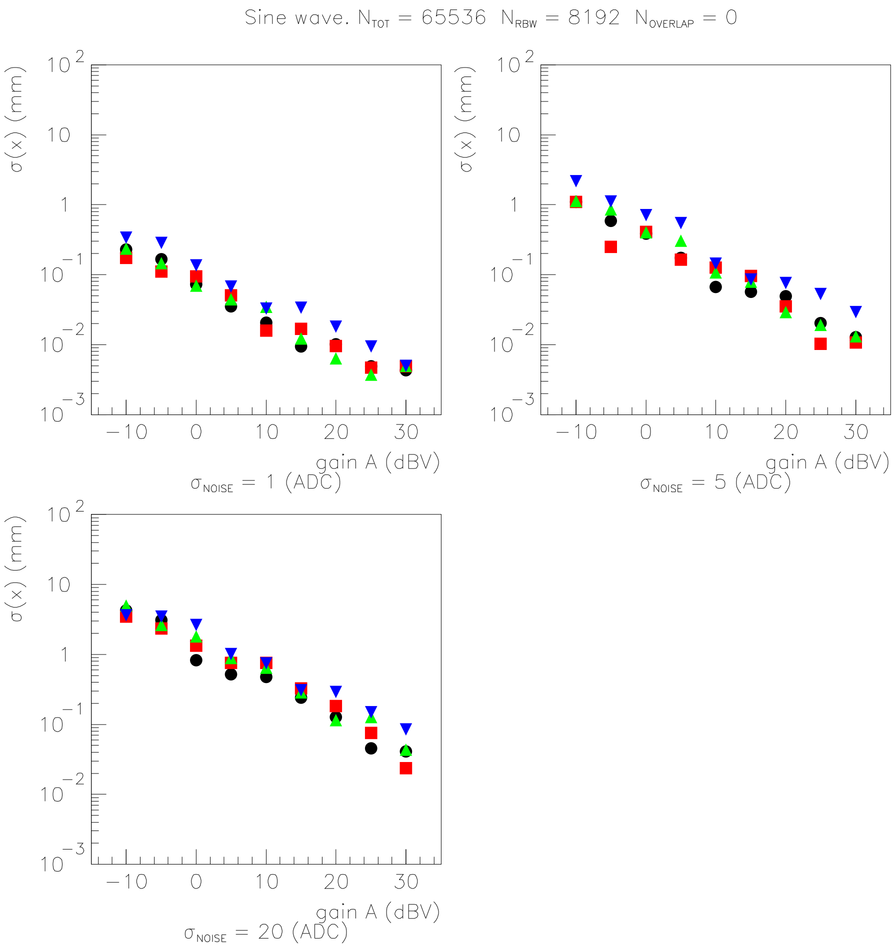

Figure 6: Standard deviation of the simulated position measurements. The generated position was 0 mm (black dots), 5 mm (red squares), 20 mm (green triangles pointing upwards) and 50 mm (blue triangles pointing downwards).

5 Memory Management

The amount of DSP memory used for the algorithm memory is limited to 192 KB. The FFT algorithm only use the first 8192 samples. This led to statistically poor measurements, with relatively large fluctuations. We decided to allocate memory for the sinusoid lookup table of 60 KB and for two ping-pong buffers with each having 60 KB. While one buffer is being analyzed, the next subset of data is transfered from the ADC's memory to the other ping-pong buffer in the DSP algorithm memory area. The memory management is done so that it is transparent to the amplitude measurement algorithm; in this way, a data sample of any length can be analyzed in its entirety. By going from 8192 to 65536 data points, for instance, the fluctuations are expected to be reduced by a factor of $\sqrt{8} = 2.83$.

6 Performance with the Calibration System

The algorithm has been tested with the real readout hardware using the calibration system. It has also been compared with the FFT algorithm.

Figures 7 and 8 show the distribution of 100 position measurements from one BPM for different intensities into the ADC, respectively. The three histograms in each figure correspond to the FFT algorithm, the heterodyne algorithm with $N_T = 65536$, $N_S = 65536$ and $N_H = 65536$, and the heterodyne algorithm with $N_T = 65536$, $N_S = 8192$ and $N_H = 4096$ (VID is just the name of the heterodyne algorithm after the addition of the video-averaging feature).

The algorithms' mean positions agree at each intensity. The rms spread decreases by better than a factor of 3.75 which is more than the expected decrease due to 8 times the statistics. For the new algorithm, the rms spread decreased from 0.04 mm to 0.01 mm when the signal intensity increased by a factor of 5.

7 Speed of Algorithms

Table 1 shows the length of time for the BPM system to complete ten consecutive measurements. The ARM and TRG clock events were set to $0F$ which means that the DSP was in the IDLE or ARM state for $\approx$ 1.0 second for the ten measurements. It takes the same amount of time for $N_T = 8192$ using the FFT and HET algorithms. The time is nearly linear with N_T when the HET algorithm is used. The VID algorithm with no overlap ($N_H = N_S$) is a little slower than the HET algorithm for $N_T = 65536$. The ratio of N_S/N_H is the factor for the increase of time to do the calculations from the no overlap case ($N_H = N_S$) due to the increased number of times that each point is used (simply more calculations are done).

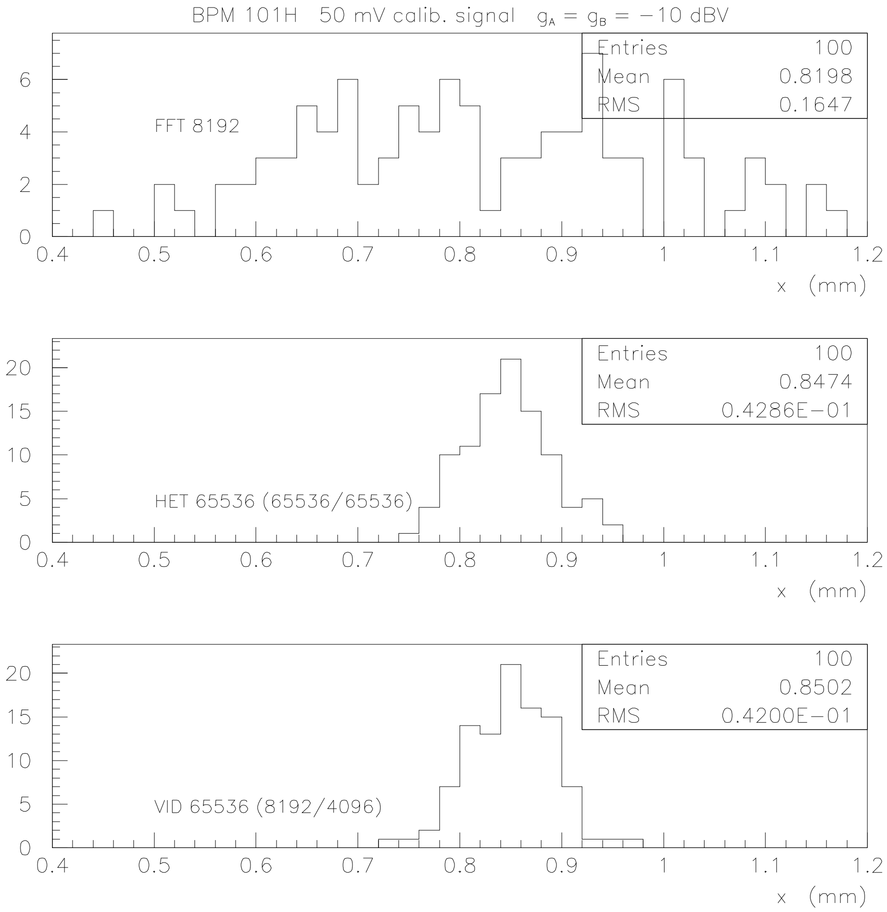

Figure 7: Calibration setup. Comparison of position measurements from three algorithms. Calibration signal of 50 mV and DC card gain set at −10 dBV.

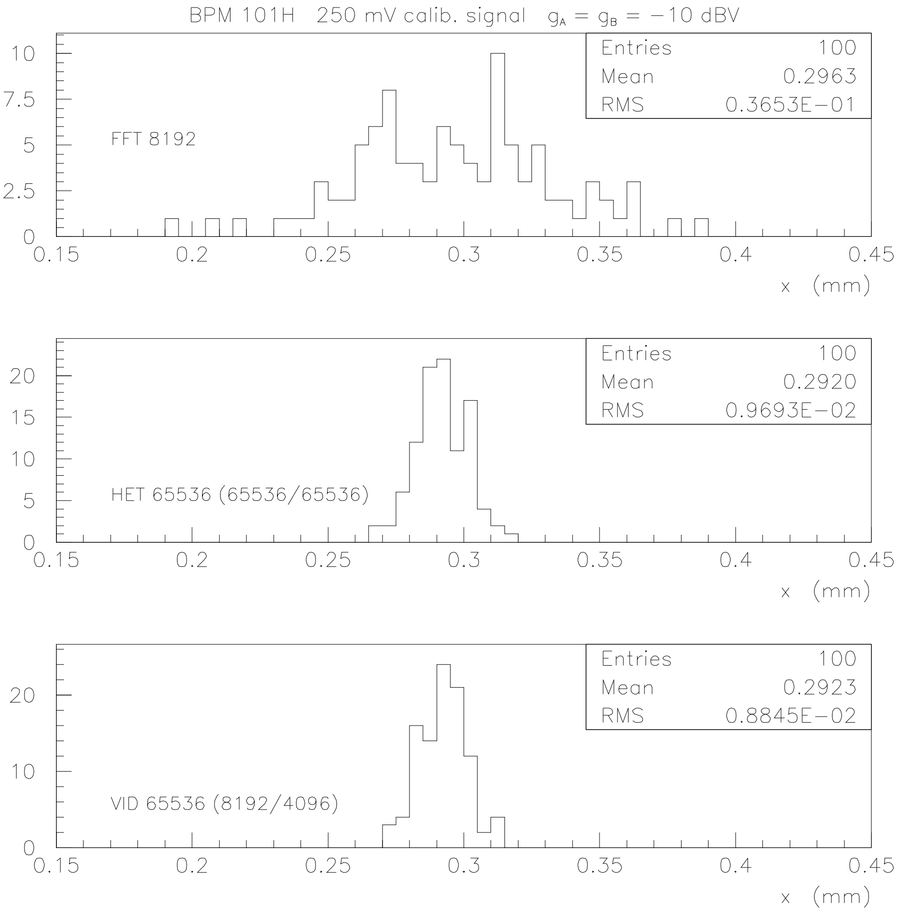

Figure 8: Calibration setup. Comparison of position measurements from three algorithms. Calibration signal of 250 mV and DC card gain set at −10 dBV.

Algorithm	Number of Points N_T	N_S	N_H	Time for 10 Measurements (s)
FFT	8192	-	-	12.5
HET	8192	8192	8192	12.4
HET	16384	16384	16384	16
HET	32768	32768	32768	32
HET	65536	65536	65536	59
VID	65536	8192	8192	62
VID	65536	8192	4096	117
VID	65536	8192	2048	240

Table 1: The time to make 10 consecutive measurements with the different algorithms and algorithm parameters.

8 Conclusion

A new algorithm has been successfully implemented with the DSPs of the BPM system. A FFT method has been replaced by a one-point Fourier transform. The new algorithm uses a ping-pong method to access the ADC data. Since the new algorithm uses more of the available ADC data, the new method is more accurate. The speed of the new and old algorithms are equivalent for the small data set that had been used. To have expanded the FFT data sample, the processing time should have increased as $N_T \log N_T$; whereas the new algorithm processing time has been shown to go as N_T. The RF is needed by the new algorithm; the more accurate the better. There is the option to analyze the data in segments smaller than the total data set in order to perform a video average. This will be helpful if the RF is not known accurately. Further tests will be performed when beam becomes available.