Design of an on-line dynamic magnetic flux analyzer based on double-buffer sliding DFT

Pasquale Arpaia[1], Lucio Fiscarelli[1,2], Carlo Petrone[1,2], Nicola Cardines[2]

[1]Department of Engineering, University of Sannio, Corso Garibaldi 107, 82100 Benevento, Italy.
Ph : +39 0824 305804-17, Fax: +39 0824 305840, E-mail: arpaia@unisannio.it
[2]Dept. of Technology, Group of Magnets Superconductors Cryostats, CERN, CH 1211, Genève 23, Switzerland.
Ph : +41 22 76 76635, Fax: +41 22 76 76230, E-mail: lucio.fiscarelli@cern.ch

Abstract- An on-line dynamic magnetic flux analyzer for magnetic measurement is presented. The instrument achieves on-line harmonic analysis by means of an improved implementation of the sliding DFT algorithm on the basis of the Fast Digital Integrator [1]. In particular, the round-off error, unacceptable in digital integrators, is corrected by employing two buffers alternatively. Details about the structure of the instrument and the algorithm implementation are pointed out, and some experimental results are analyzed in order to validate the proposal.

I. Introduction

At CERN, after the qualification effort of the Large Hadron Collider's magnets, a new generation of rotating coils [2] requests digital integrators with an increased flux sampling rate (100 kS/s) and a higher accuracy (10 ppm) [3]. In cooperation with the University of Sannio (Italy), during the last years, the Fast Digital Integrator (FDI) has been proposed for a real-time on-board batch integration of rotating coil signals [1] . This instrument can analyse the magnetic flux over a bandwidth larger than state-of-art solutions [4]-[5] with higher accuracy owing to its 18 bit-resolution and 500 kS/s-rate digital conversion. A custom programmable gain amplifier (PGA), and an analog-to-digital converter (ADC) are the core of the FDI measurement machine. Moreover, it allows angular and time domain to be linked within a resolution of 50 ns by means of an absolute time base. The DSP is the main board processor, interacting with the measurement machine and the external PXI bus through a field programmable gate array (FPGA).

Starting from this well-characterized hardware layer, a new firmware scheme has been developed in order to allow the on-line computation of a selected set of interesting magnetic flux harmonic coefficients. Typically in magnetic measurements, a voltage signal, generated by the interaction of a calibrated coil and the magnetic field, is integrated in order to obtain the flux signal. Then, the flux is analysed to compute the magnetic field intensity in the region of action of the coil. In particular, in the case of rotating coils [2], widely used in magnet characterization, the analysis of the flux is based on the Fourier transform for computing the harmonic coefficients which are related to the geometric distribution of magnetic field. Usually the signal coming from the coils is acquired by a high resolution acquisition system and then processed off-line. In case of measurements in magnets fed with variable current, the coils have to turn at high speed compared to the current ramp rate in order to minimize the error due to the field variation over coil rotations, with the drawback that a huge amount of data has to be transferred, stored and analysed in a short time interval. With the new generation of hardware for magnetic measurements [5]-[6], often DSP-based systems, a first step of the data analysis is carried out on-line on the acquisition board and the integral over a certain time interval is given in output instead of the values of all the voltage samples. For this aim, the sliding DFT algorithm is exploited due to its low computation complexity in case only few harmonics are needed. Nevertheless, the computing rule of the algorithm implies an unavoidable rounding error when applied on a finite precision processor machine. This error could be particularly disadvantageous in case of long measurements like the complete dynamic characterization of magnets.

In this paper, the rounding error problem is overcome by restarting the calculation after a predetermined number of iterations. In order to avoid losses in the measurement data, two buffers for current values of the coefficients are used and reset alternatively, with a little surcharge of memory resources.

In the following, in Section 2, the architecture of the digital integrator and the algorithm structure are shown, and in Section 3, some experimental results are illustrated and discussed.

II. Proposal

The proposed dynamic magnetic flux analyzer (DFA) shifts another step of the analysis on board: by exploiting the DSP computation capability and a specific algorithm, the harmonics related to the acquired flux are computed on-line and only the required ones are transferred from the board to the pc-based processing and storage unit, with the main advantage of having a data flow reduced by a factor 1,000 typically compared to a standard acquisition board.

A. The instrument architecture

In the proposed instrument architecture (Figure 1), the input signal arising from the coil is conditioned by a low-noise custom differential programmable gain amplifier (PGA). It is provided by an automatic procedure for calibration and correction of gain and offset errors. A field-programmable gate array acts as an input–output processor: it supervises the PGA operations at low level and provides the interface for the board bus peripheral component interface extensions for instrumentation (PXI). A DSP supervises the board as a whole and integrates online the data digitized by the 18-bit 500-kS/s differential ADC by exploiting the trigger pulses of an encoder mounted on the shaft of the rotating coil. These pulses, coming out from the flux sampler, are measured by a time base with a resolution of 50 ns.

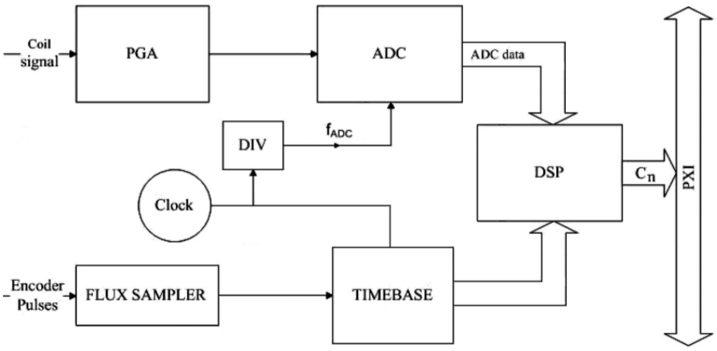


Figure 1. The instrument architecture.

B. The computing algorithm based on the sliding DFT

Probably the most frequently operated task in modern computers in any way is the Discrete Fourier Transform (DFT). In signal processing the DFT is used to carry out several kinds of analysis in discrete time signals sequences [7]. Of the numerous available algorithms, the Cooley-Tukey Algorithm (FFT) [8] is the most commonly used.

If N designates the number of samples the FFT is supposed to take into account, the complexity *O(N)* may be expressed with N $\log_2$ N as the result of a divide and conquer strategy. Compared to the complexity $O(N^2)$ of a non-optimized DFT, the gain is considerable. However the disadvantage for on-line applications is that the FFT must be applied to a whole N-sized dataset, where N must be a power of 2. If no channel switching procedure is added, the FFT harmonics are updated at maximum rate of $f = f_s/N$, where f_s is the sampling frequency. Moreover, to release the harmonic values related to one dataset just after the last sample is available, the computation effort has to be carried out in a very little time, a huge drawback for DSP-based application.

For the proposed aim, a more appropriate algorithm should be the sliding Discrete Fourier Transform (sDFT) [9] [10]. The sliding discrete Fourier transform (sDFT) algorithm performs a N-point DFT on time samples within a sliding-window. Basically, the sDFT is based on a simple consideration: for two successive time instants, *n-1* and *n*, the windowed sequences *x(n-1)* and *x(n)* contain essentially the same elements apart from one.

The sDFT algorithm begins by computing the DFT of N time samples, the time window is then advanced one sample, and a new N-point DFT is calculated. The value of this process is that each new DFT is efficiently computed directly from the results of the previous DFT. The incremental advance of the time window for each output computation leads the name sliding DFT or sliding-window DFT.

The principle used to obtain the time-domain difference equation is known as the DFT shifting theorem or the circular shift property. It states that if the DFT of a windowed (finite-length) time-domain sequence is *x(n)*, then the generic DFT output X_k, circularly shifted by one sample to left, is $\mathrm{X_k(n)e^{i2\pi k/N}}$.

Hence we can express the kth bin by the following time-domain difference equation:

$$\mathrm{X_k(n) = [X_k(n-1) - x(n-N) + x(n)]\, e^{i2\pi k/N}} \qquad (1)$$

where $X_k(n)$ is the new spectral component and X_k *(n-1)* is the spectral component evaluated in the previous step.
Equation (1) reveals the value of this process in computing real-time spectra. The sDFT requires only one complex multiplication and two real additions per harmonic sample if it is already known for the previous time instant. The computational complexity of each time successive N-bins output is then *O(N)* for the sliding DFT compared to $O(N^2)$ of the DFT and $O(Nlog_2N)$ of the FFT. Unlike the DFT or FFT, however, due to its recursive nature, the sliding DFT output must be computed for each new input sample. When the harmonic coefficients are required every M input samples, and M is less than $\log_2$ (N), the sliding DFT can be computationally better than traditional FFT implementations even when all the harmonic coefficients are required. If a new N-output DFT output is required only every N inputs, the sliding DFT requires $O(N^2)$ computations and is equivalent to the DFT.

From (1), in the classical Z transform representation, the single-bin sDFT filter structure is shown in Figure 2. The single-bin calculus is implemented as an IIR filter with a comb filter followed by a complex resonator. The comb filter delay of N samples forces the filter transient response to be N-1 samples in length, so the output will not reach steady state until the X_k *(N)* coefficient is available.

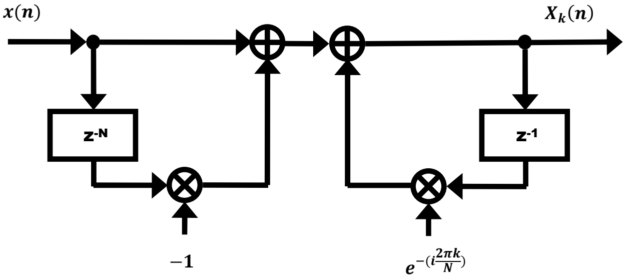


Figure 2. Single-bin sDFT filter structure.

The z-domain transfer function for the kth bin of the sliding DFT filter is:

$$\mathrm{H_{SDFT}(z)} = \frac{(1-z^{-N})}{1-e^{i2\pi k/N}\,z^{-1}} \qquad (2)$$

In practical applications, the algorithm can be initialized with zero input (*x(0)=0*) and zero output ($X_k(0)=0$). Obviously, the output will not be valid until N input samples have been processed.

One of the attributes of the sDFT is that once X_k *(n-1)* is obtained, the number of computations to calculate $X_k(n)$ is fixed and independent of N. Unlike the radix-2 FFT, in sDFT algorithm N can be any positive integer rather than a power of two, giving us greater flexibility to tune the central frequency by defining N such $N=l\cdot f_s/f_i$ where f_s is the sampling frequency, f_i the frequency of interest and *l* an integer factor.

C. Round-off error and the "double buffer" correction

The algorithm based on the formula (1), when applied on a finite precision processor, like DSP, implies an unavoidable rounding error [11]-[12]. In Fig. 3, the error on the first harmonic coefficient, computed in simulation on a synthetic signal containing only the first harmonic ($f/f_{sampling}=1024$) and with a 1024 points window, relative to the actual value, is depicted. This error is due mainly to the sum of very different magnitude floating-point values and it could be particularly disadvantageous in case of long measurements because of the accumulation nature of the error at each step.

In this proposal, the rounding error is overcome by restarting the calculation after a predetermined number of iterations. In order to avoid loss in the output data, two independent sets (from that the name "double buffer") of current harmonics values are computed. The output data are then taken from the oldest reset buffer. This implementation involves a little surcharge of memory since the harmonics have to be stored twice; also in this case the algorithm has a better behaviour in case that few harmonics are needed.

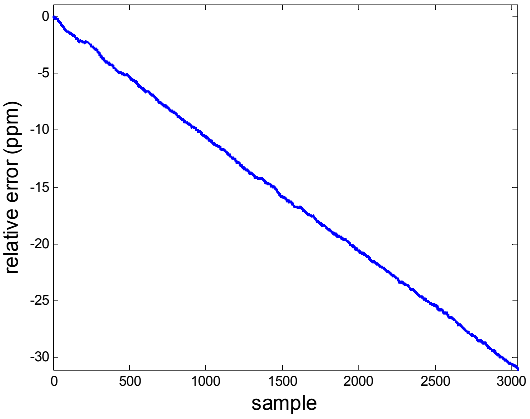
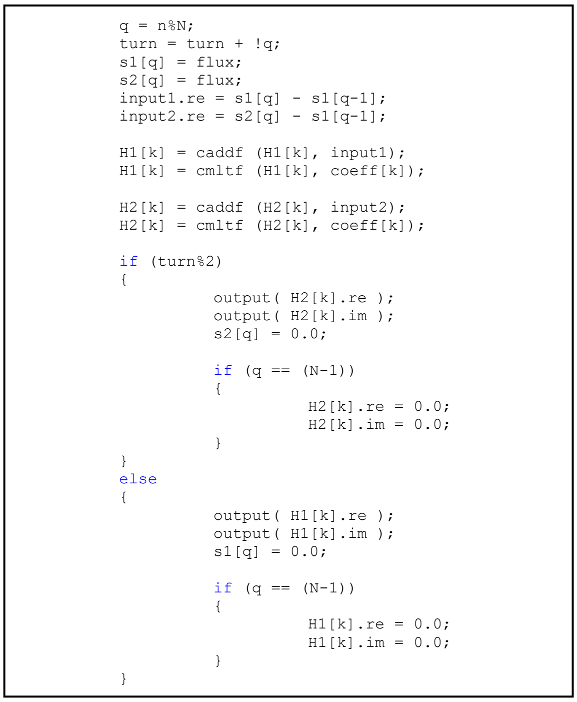


Figure 3. Round-off error in sDFT standard algorithm.

The implementation in C code of one bin sDFT double buffer algorithm is shown in figure 4: *s1* and *s2* are two circular buffers containing the flux samples; *caddf()* and *cmltf()* are functions for computing respectively the complex addition and multiplication; *H1* and *H2* are the buffers for the harmonics; and *output()* is the function for throwing out the values.



```
q = n%N;
turn = turn + !q;
s1[q] = flux;
s2[q] = flux;
input1.re = s1[q] - s1[q-1];
input2.re = s2[q] - s1[q-1];

H1[k] = caddf (H1[k], input1);
H1[k] = cmltf (H1[k], coeff[k]);

H2[k] = caddf (H2[k], input2);
H2[k] = cmltf (H2[k], coeff[k]);

if (turn%2)
{
        output( H2[k].re );
        output( H2[k].im );
        s2[q] = 0.0;

        if (q == (N-1))
        {
                H2[k].re = 0.0;
                H2[k].im = 0.0;
        }
}
else
{
        output( H1[k].re );
        output( H1[k].im );
        s1[q] = 0.0;

        if (q == (N-1))
        {
                H1[k].re = 0.0;
                H1[k].im = 0.0;
        }
}
```

Figure 4. Code implementation of the kth bin sDFT with double buffer correction.

In Fig. 5, a comparison of the relative error on the first harmonic coefficient of the sDFT standard algorithm and the “double buffer” version is shown. The error of the “double buffer” algorithm is delimited and reset to zero every N points, where N is the length of the observing window, while the error of the standard sDFT implementation grows indefinitely.

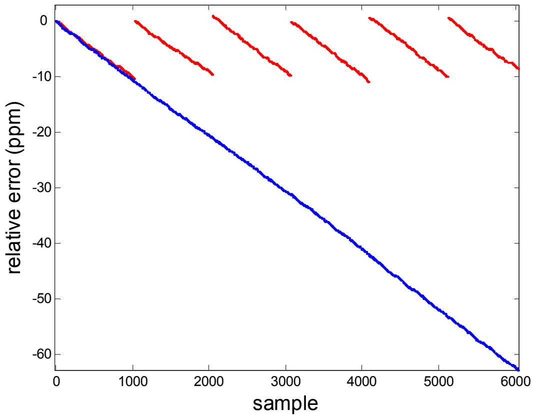


Figure 5. Simulation relative errors comparison, sDFT standard in blue and “double buffer” in red.

III. Experimental results

The proposed algorithm was integrated in the existing firmware of an FDI board [1], in order to implement the DFA feature, and some tests were carried out.

In Fig. 6, the test set-up is shown: a PXI crate containing the DFA board is linked to a PC, a Stanford DS360 low noise function generator is used as signal source, and a TTI TG1010 function generator is also exploited as trigger generator.

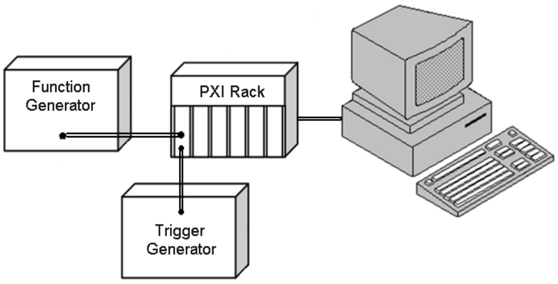


Figure 6. Test set-up.

A sinusoidal signal of 5 V and 1 Hz is acquired by the DFA board triggered with a signal of 1024 Hz, the resulting harmonics were transferred through the PXI bus on the PC and then analysed.

The results, depicted in Figure 7, show the foreseen behaviour of the error: it remains limited up 20 ppm independently from the test duration.

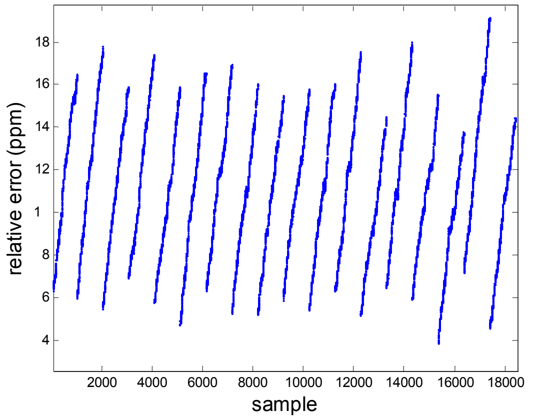


Figure 7. Relative error of the experimental test.

Conclusions

On the basis of the previously developed Fast Digital Integrator [1], a Dynamic Flux Analyser is proposed. The hardware and firmware of the new measurement instrument have been designed and implemented. In particular, a specific computing algorithm, derived from a sliding DFT with a correction mechanism for making stable the round-off error, has been developed and preliminary results, from numerical simulations and experimental tests, have been analysed. The future work will be spent to validate the instrument on the field with specific case studies.

Acknowledgments

The Authors thank Luca Bottura, Felice Cennamo, and Louis Walckiers for the useful suggestions.

References

[1] P. Arpaia, A. Masi, G. Spiezia, “A Digital Integrator for Fast and Accurate Measurement of Magnetic Flux by Rotating Coils”, IEEE Transactions on Instrumentation and Measurement, Vol. 56, No. 2, April 2007.

[2] L. Bottura, K. N. Henrichsen, “Field Measurements”, CERN Accelerator School Proceedings, CERN, September 2004, pp. 118-151.

[3] S. Amet, L. Bottura, L. Deniau, and L. Walckiers, “The multipoles factory: an element of the LHC control”, IEEE Trans. Appl. Supercond. Vol. 12, pp. 1417-1421, 2002.

[4] P. Galbraith, “Portable Digital integrator”, 1993, CERN Internal Technical Note 93-50, AT-MA/PF/fm

[5] R. Carcagno, J. DiMarco, S. Kotelnikov, M. Lamm, A. Makulski, V. Maroussov, R. Nehring, J. Nogiec, D. Orris, O. Poukhov, F. Prakoshin, P. Schlabach, J.C. Tompkins, G.V. Velev, “A fast continuous magnetic field measurement system based on digital signal processor”, Proc. of 19th Magnet Technology Conference, Genoa, 18-23 September 2005.

[6] P. Arpaia, J. G. Perez, A. Masi, G. Spiezia, “Metrological characterization of a Fast Digital Integrator for magnetic measurement at CERN”, Proc. of XVIII IMEKO World Conf., Sep., 17 – 22, 2006, Rio de Janeiro, Brazil.

[7] J. Proakis, G., D., M. Manalakis, "Digital Signal processing, Principles. Algorithms, and Applications", Third Edition, Prentice Hall lnc., Engelwood Cliffs, New Jersey 1996.

[8] J. W. Cooley and J. W. Tukey, An algorithm for machine calculation of complex Fourier series, Math. Comp., 19 (1965), 297-301.

[9] E. Jacobsen, R. Lyons, “The Sliding DFT", IEEE Signal Processing Magazine, Mar. 2003, 74-80.

[10] E. Jacobsen and R. Lyons. “An update to the Sliding DFT", IEEE Signal Processing Magazine, Jan. 2004, pp.110-111.

[11] T. Fujii, H. Harashima, H.Miyakawa, “Synthesis of sliding discrete Fourier transform and sliding Hadamard transform circuits”, Electronics and Communications in Japan, Vol. 64-A, No. 12, 1981.

[12] S.C. Douglas, J.K. Soh, “A numerically-stable sliding-window estimator and its application to adaptive filters”, Signals Systems & Computers, 1997, 111 - 115 vol.1.