A resource-efficient adaptive Fourier analyzer

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2016 JINST 11 P10014

(http://iopscience.iop.org/1748-0221/11/10/P10014)

View the table of contents for this issue, or go to the journal homepage for more

Download details:

IP Address: 128.141.192.151
This content was downloaded on 20/03/2017 at 12:22

Please note that terms and conditions apply.

You may also be interested in:

STUDY OF EXOTIC WEAKLY BOUND NUCLEI USING MAGNETIC ANALYZER MAVR
V.A. Maslov, V.I. Kazacha, I.V. Kolesov et al.

Vibration damping with negative capacitance shunts: theory and experiment
B de Marneffe and A Preumont

Contrast Adaptation Decreases Complexity in Retinal GanglionCell Spike Train
Wang Guang-Li, Huang Shi-Yong, Zhang Ying-Ying et al.

L D Bakhrakh

Energy mitigation, adaptation and biodiversity: Synergies and antagonisms
P M Berry and J S Paterson

Manufacture and characterization of ionic polymer transducers
M D Bennett and D J Leo

How will climate change affect spatial planning in agricultural and natural environments? Examples from three Dutch case study regions
Margaretha Blom-Zandstra, Maurice Paulissen, Herman Agricola et al.

Analyzers of charged particles flows with different functional capabilities
A O Saulebekov and Zh T Kambarova
A resource-efficient adaptive Fourier analyzer

C.F. Hajdu, a,b,1 C. Zamantzas a and T. Dabóczi b

a Beams Department, European Organisation for Nuclear Research (CERN),
385 Route de Meyrin, Meyrin, CH-1217 Switzerland

b Department of Measurement and Information Systems, Budapest University of Technology and Economics,
Műegyetem rkp. 3, Budapest, H-1111 Hungary

E-mail: Csaba.F.Hajdu@cern.ch

ABSTRACT: We present a resource-efficient frequency adaptation method to complement the Fourier analyzer proposed by Péceli. The novel frequency adaptation scheme is based on the adaptive Fourier analyzer suggested by Nagy. The frequency adaptation method was elaborated with a view to realizing a detector connectivity check on an FPGA in a new beam loss monitoring (BLM) system, currently being developed for beam setup and machine protection of the particle accelerators at the European Organisation for Nuclear Research (CERN). The paper summarizes the Fourier analyzer to the extent relevant to this work and the basic principle of the related frequency adaptation methods. It then outlines the suggested new scheme, presents practical considerations for implementing it and underpins it with an example and the corresponding operational experience.

KEYWORDS: Control and monitor systems online; Data processing methods; Detector control systems (detector and experiment monitoring and slow-control systems, architecture, hardware, algorithms, databases); Digital signal processing (DSP)

1 Corresponding author.
1 Introduction

Recursive estimation methods based on signal models allow the measurement of various parameters or the calculation of different transforms of signals. Their recursive nature renders them particularly suitable for real-time applications.

These methods consider the signal under measurement as the output of a hypothetical dynamic system: the signal model. The procedure then consists of estimating the state vector of this system. Measurement systems based on a deterministic signal model are referred to as observers.

Among the observer-based approaches, Hostetter suggested a recursive algorithm to calculate the discrete Fourier transform of a signal [1]. Péceli extended this method to allow calculating arbitrary linear signal transforms recursively [2]. When Péceli’s resonator-based observer structure is used as a spectral observer, it is referred to as Fourier analyzer (FA).

These algorithms rely on a linear system model with known parameters. In the case of spectral analysis, however, the frequencies of the harmonic components of the signal may be unknown. Nagy proposed a way of implementing frequency adaptation in the FA. This method, called adaptive Fourier analyzer (AFA), allows locking the resonator structure to the fundamental frequency of the signal in a PLL-like manner, following slow frequency drifts [3, 4]. It has been used successfully in several industrial applications, such as a vector voltmeter [3, 4], and active noise and distortion cancellation [5, 6].
Several modifications to this procedure have been put forward. These frequency adaptation methods, outlined in section 2.2, all require calculating phase differences between complex numbers. Since the complex calculations required by the Fourier analyzer are best carried out in the algebraic form \((z = a + jb)\), phase calculations become computationally intensive. In this paper, we describe a new frequency adaptation scheme avoiding phase calculations at the expense of slower adaptation of the fundamental harmonic frequency, while potentially reducing sensitivity to noise.

Section 2 of this paper describes the Fourier analyzer and the basic principle of the frequency adaptation method along with an enumeration of the modifications that have been suggested. Section 3 introduces the new frequency adaptation scheme, while section 4 presents the application which motivated its elaboration, along with practical considerations for the implementation. Section 5 then concludes the paper.

2 The resonator-based observer

2.1 The Fourier analyzer

Péceli’s resonator-based observer estimates the state variables of the conceptual signal model [2]. The Fourier analyzer is obtained when using this structure as a spectral observer.

The state variables \(x_{i,n}\) of the signal model shown in figure 1 correspond to the complex Fourier coefficients of the signal, therefore, the signal model can be interpreted as a complex multisine generator reconstructing the signal from its discrete Fourier transform (DFT), with each of its state variables (channels) representing a harmonic resonator of the corresponding frequency.

We are assuming the signal to be real-valued. In this case, the conceptual signal model is described by the following equations:

\[
\begin{align*}
x_{n+1} &= x_n, \\
x_n &= [x_{i,n}]^T \in \mathbb{C}^{N \times 1}, \quad i = -K, \ldots, K, \\
y_n &= c_n^T x_n,
\end{align*}
\]
Figure 2. Block diagram of the resonator-based observer. Notice how the structure of the observer replicates that of the conceptual signal model shown in Figure 1. It is extended by the time-varying $g_{i,n}$ coefficients at the input of each integrator and the common feedback.

The system equations of the observer are as follows:

$$\hat{x}_{n+1} = \hat{x}_n + g \hat{e}_n = \hat{x}_n + g (y_n - \hat{y}_n), \quad (2.6)$$

$$\hat{x}_n = [\hat{x}_{i,n}]^T \in \mathbb{C}^{N \times 1}, \quad i = -K, \ldots, K, \quad (2.7)$$

$$\hat{y}_n = c_n^T \hat{x}_n, \quad (2.8)$$

$$g_n = [g_{i,n}]^T \in \mathbb{C}^{N \times 1}, \quad i = -K, \ldots, K, \quad (2.9)$$

$$g_{i,n} = \frac{\alpha}{N} c_{i,n}^*, \quad (2.10)$$

where $\hat{x}_n$ is the estimated state vector, $\hat{y}_n$ is the estimated signal and $e_n$ is the estimation error. The vector $g_n$ is the product of $\frac{\alpha}{N}$, a freely tunable parameter referred to as observer gain determining the dynamical behavior of the observer, and $c_n$, a vector of complex roots of unity setting the poles and thereby the frequency response of the individual observer channels. If the vectors $c_n$ and $g_n$ are set according to (2.5) and (2.10), respectively, with $\alpha = 1$ and $f_1 = \frac{f_0}{2}$, a deadbeat observer is obtained which settles in at most $N$ steps and calculates the recursive discrete Fourier transform of the input signal [7].
The transfer function of channel $i$ in the feedback loop from $y_n$ to $\hat{x}_{i,n} \cdot c_{i,n}$ (from the input of the structure to the adder at the output, see figure 2) can be expressed as [8]:

$$H_i(z) = \frac{r_i\bar{z}_{i1}}{1 + \sum_{m=-K}^{K} \frac{r_m\bar{z}_{m0}}{z^{-m}}},$$

(2.11)

where $r_i = \frac{1}{N}, \forall i$ in case of the Fourier analyzer.

### 2.2 Frequency adaptation

If the fundamental frequencies of the signal and the Fourier analyzer are different, Nagy suggested a method to estimate the frequency difference from the state variables of the observer and to adapt the frequencies of the resonators to match the signal [3].

The input signal of the observer can be expressed in the following form, as reconstructed from its Fourier series, with $a_i$ corresponding to its complex Fourier coefficients:

$$y_n = \sum_{i=-L}^{L} a_i e^{j2\pi f_i n}.$$  

(2.12)

At present, a difference between the fundamental frequencies of the signal and the Fourier analyzer is assumed. They will be denoted by $f_0$ and $f_r$ relative to the sampling frequency, respectively. Accordingly, the signal doesn’t necessarily have the same number of harmonics as the number of resonators in the observer.

If $f_0 = f_r$, then after settling, the estimators $\hat{x}_i$ are equal to the Fourier coefficients $a_i$. Otherwise, the estimator of the fundamental harmonic in the $n$th time step can be expressed as follows [8]:

$$\hat{x}_{1,n} = \sum_{i=-L}^{L} a_i e^{j2\pi f_i n} H_1\left(e^{j2\pi f_i}\right) c_{1,n}^{-1} = \sum_{i=-L}^{L} a'_i e^{j2\pi (f_i - f_r) n},$$  

(2.13)

where $\hat{x}_{1,i,n}$ is the contribution from the $i$th harmonic component of the signal in the $n$th time step to the estimator of the fundamental harmonic. The transfer function $H_1\left(e^{j2\pi f_i}\right)$ of the fundamental harmonic resonator from $y_n$ to $\hat{x}_{1,n} \cdot c_{1,n}$ at the corresponding frequency was defined in (2.11).

From (2.13), it follows that $\arg \hat{x}_{1,i,n+1} - \arg \hat{x}_{1,i,n} = 2\pi (f_0 - f_r)$. If $|a'_i| \ll |a_i^0| \ (i \neq 1)$, then $\hat{x}_{1,n}$ provides a good estimate of $\hat{x}_{1,i,n}$ and the error in the frequency of the fundamental resonator relative to the sampling frequency can be expressed as:

$$f_0 - f_r \approx \frac{\arg \hat{x}_{1,i,n+1} - \arg \hat{x}_{1,i,n}}{2\pi}. $$

(2.14)

Equation (2.14) can be used as a basis for adapting the resonator frequencies, and the principle of the frequency adaptation scheme stems from it. In the adaptive Fourier analyzer (AFA), Nagy suggested updating $f_r$ by $\frac{1}{N} \cdot (f_0 - f_r)$ as obtained through (2.14) in every time step. If the change in the frequency of the base harmonic warrants it, the number of resonator pairs $K$ must also be
updated so that $K \cdot f_1 < \frac{f_s}{2} \leq (K + 1) \cdot f_1$ still holds, in order to ensure a uniform distribution of the resonator positions up to $\frac{f_s}{2}$ [3].

Several modifications to this procedure have been put forward. Applying a damping factor to the frequency update has been suggested in order to improve noise rejection (robust AFA, RA). Simon and Péceli developed a block-wise frequency adaptation method (block AFA, BAFA) in order to evaluate its convergence properties [8]. With a view to decreasing sensitivity to noise, Ronk suggested extending these approaches by averaging the Fourier components used for the adaptation, thus obtaining the eBAFA from the BAFA [9], and the improved robust AFA (IRA) from the RA [10]. Dabóczi then proposed combining the IRA and the BAFA, yielding the extended improved robust AFA (eIRA) [11].

3 The zero crossing adaptation scheme

The frequency adaptation methods referred to earlier all rely on calculating phase differences between differently spaced samples of the estimator of the fundamental harmonic, then approximating the frequency difference through numerical manipulation.

We propose estimating the frequency mismatch by tracking when the phase of the fundamental harmonic, $\arg \hat{x}_{1,n}$, crosses zero. The phase difference between two consecutive zero crossings is $2\pi$, thus if $N_z$ zero crossings are detected spanning a time interval $t_m$, an approximation of the frequency error relative to the sampling frequency can be calculated as follows based on (2.14):

$$|f_0 - f_r| \approx \frac{(N_z - 1) \cdot 2\pi}{2\pi} = \frac{N_z - 1}{t_m \cdot f_s}, \quad (3.1)$$

where the sign of the frequency difference can be determined based on the direction of the zero crossings in phase, as described in section 4.5. The new value of $f_r$ can then be used for adapting the frequency of the resonators.

This method of calculating the frequency mismatch presents the following advantageous characteristics:

- Detecting zero crossings in phase is very efficient in terms of FPGA resources for most practically feasible implementations. For complex numbers represented in polar form ($z = A e^{i\phi}$), finding changes in the sign bit of the phase is sufficient. In case the algebraic form ($z = a + jb$) is used, zero crossings in phase can be identified by considering the signs of the real and imaginary parts simultaneously.

- In case the observed signal and thus the resonator phase are subject to noise, this approach may allow reducing the resulting influence on the estimator of the frequency, as illustrated in section 4.7.

4 Detector connectivity checks with the Fourier analyzer

4.1 Motivation

Beam loss monitoring (BLM) is an essential method for the protection of particle accelerators. It consists in measuring the showers of secondary particles generated by the primaries escaping from the particle beams under acceleration, impacting on an aperture limit of the machine. BLM systems
are the backbone of the strategy for beam setup and machine protection of all particle accelerators at the European Organisation for Nuclear Research (CERN), one of the world’s leading particle physics research facilities.

The measurement data acquired by the BLM system are processed in real time and if required for protecting the machines from excessive energy deposition by the beam, a decision maker process running continuously is able to trigger the safe extraction of circulating beams and the inhibition of further injections. Specifically, the rapid response of the BLM system is critical for protecting the machines from short and intense particle losses [12]. Moreover, subsets of the measurement data are displayed in the control room to assist the operators and archived for machine tuning and calibration.

Continuous supervision of the connectivity and correct operation of the BLM detectors, and the functionality of the entire BLM system is therefore essential, yet to our knowledge, no particle accelerator in the world has this feature at present. The current best solution is in operation at CERN’s flagship machine, the Large Hadron Collider (LHC), where a connectivity check requiring the accelerator to be offline is enforced on each detector channel every 24 hours. This procedure relies on modulating the bias high voltage of the ionization chambers used as detectors and detecting the corresponding modulation generated in their output current [13, 14].

Within the framework of a major accelerator upgrade project at CERN, a new BLM system for the preaccelerators of the LHC is in an advanced state of development. This complex of accelerators is referred to as LHC injectors. In the new BLM system, it is foreseen to use ionization chambers similar to those in the LHC system in most cases, however, other detector types such as diamond detectors, secondary emission monitors and Cherenkov detectors will be installed in some specific locations [15].

Developing a new process continuously supervising the functionality of the BLM system for the injectors is mandatory. Contrary to the LHC, where the cycle of the machine contains long periods without beam while the new fill is being prepared, the operation of the injectors is essentially uninterrupted. This means that any such process needs to run in parallel to the data acquisition without interfering with it.

In the digitized output signal of the ionization chambers, the switching frequency of about 30 kHz of the high voltage power supply (HVPS) and its harmonics have been identified as parasitic components. Detecting these harmonics might allow realizing a non-destructive connectivity check, a highly appealing way of implementing detector supervision without external intervention by simply post-processing the digitized output signal.

4.2 The suggested solution

In the BLM system under development for the injectors, each front-end card acquires and digitizes the output current of eight detectors at 500 ksp, using an innovative measurement technique featuring a fully differential integrator [16]. The data is then forwarded to the corresponding back-end card over a high-bandwidth optical link for further processing. In order to ensure high throughput and flexibility, the treatment of the digitized data on these cards is based on reconfigurable FPGA devices. This offers a convenient way to implement additional data processing. For the on-line connectivity check, we chose the Altera Cyclone IV FPGA of the back-end module.
A highly customized version of Péceli’s Fourier analyzer [2] was chosen for detecting the HVPS harmonics. It’s inherently stable due to the unit negative feedback in the observer, it offers flexibility in locating the resonator poles [17] and a straightforward frequency adaptation method has been proposed to complement it [3], making it possible to track the slowly varying frequency of the HVPS contributions in the signal. The same structure could also be used to detect a low-frequency modulation if desired.

4.3 Practical considerations for the Fourier analyzer

We are implementing the resonator-based observer on an FPGA. This means that additions and multiplications have a moderate hardware cost, while divisions and angle calculations are more expensive and thus are better avoided in order to save resources.

According to (2.6)–(2.10) describing the operation of the resonator-based observer, the state update requires additions and multiplications of complex numbers. Using the polar form of complex numbers would be computationally advantageous for multiplication, however, conversions to and from the algebraic form would be required for the additions. Nevertheless, complex multiplications can be realized by a series of multiplications and additions in algebraic form, avoiding the additional FPGA resource use that would result from the conversions between the two forms. Therefore, we implemented our project using this form. We also exploited the fact that the input signal of the observer is real-valued: this implies that the resonators form complex conjugate pairs, thus realizing one of the pair is sufficient and the summed output of the two can be obtained as $2 \cdot \text{Re}\{\cdot\}$.

We considered several different solutions for the arithmetic used in the processing, since the values are expected to have a high dynamic range. In MATLAB, we implemented and tested a block floating point architecture based on fixed point arithmetic with a common but variable scaling factor $2^n$ ($n \in \mathbb{Z}$) for all fixed point numbers. However, we discontinued this approach due to its complexity. We first realized the resonator structure on an FPGA with IEEE-754 compliant 32-bit floating point arithmetic using IP cores provided by the manufacturer. In order to reduce FPGA resource use, we then converted this realization to fixed point arithmetic. Our tests showed that at least 26 fractional bits are required for satisfactory performance with data samples acquired from the prototype system when using fixed point arithmetic.

The real and imaginary parts of each state variable and coefficient are treated separately. According to (2.4) and (2.5), the $c_n$ coefficients are calculated from their initial values by exponentiation. This calculation can become numerically unstable. Since the $c_{i,n}$ coefficients are unit complex numbers, we circumvented this problem by calculating their phase instead, which can be done by simple addition. Then, the real and imaginary parts of the coefficient required by the processing are obtained in every iteration using memory-based lookup tables storing values of the cosine and sine functions. If required, the precision of this method can be improved by interpolating the values obtained from the lookup tables, but we deemed this unnecessary in our implementation.

We implemented the Fourier analyzer on an FPGA with one single processing channel, which calculates the real and imaginary parts of the new state variable for each resonator sequentially and then the error of the estimator. This processing channel consists of two multipliers and two adders, along with memory blocks storing the resonator values, the phase of the $c_n$ coefficients and the lookup tables. The operations related to the state update are parallelizable, which provides a
Figure 3. Transfer function of a single resonator channel from $y$ to $\hat{x}$, for different values of $\alpha$. The center frequency of the channel is $f_{rel} = 0.06$, that is, 30 kHz with $f_s = 500$ kHz. This implies $N = 17$. The zeroes of the transfer function correspond to the positions of the other resonators.

margin for tuning speed versus resource usage (see section 4.7 for indicative numbers). This single-channel implementation minimizes FPGA resource use, but if desired, the processing channel could be repeated to parallelize computations and increase throughput, thus the possible number of resonators.

4.4 Tailoring the Fourier analyzer to the application

The output current of the detectors is acquired at a sampling rate of 500 ksp, and since the goal is detecting a signal around 30 kHz and its harmonics, there is little margin in decreasing the sampling rate. In order to ensure continuous, on-line data processing, the observer operates at 500 kHz. If the observer were used for calculating a recursive DFT, with a resolution on the order of 10–100 Hz ensuring proper selectivity, this would imply a prohibitively high number of resonators in the entire Fourier analyzer structure realized on the FPGA we have at hand in our system: in case of a serial implementation, the processing wouldn’t be able to cope with the input data flow, and in case of a parallelized implementation, the FPGA resource cost would be too high for the design to fit. We then considered a structure with an extremely coarse frequency resolution, with the DFT points positioned at the HVPS frequency and its harmonics. At the expense of slower convergence, the $\frac{\alpha}{N}$ parameter in (2.10) can be used to control the bandwidth of the resonator channels. As shown in figure 3, this can ensure an appropriate frequency selectivity, without impacting on the stability of the processing, while reducing noise in the acquisition [18].

We then turned some of the individual resonator positions into resonator triplets. This means that for some of the lowest order harmonic resonators, we established two additional resonator positions on either side of each of the already existing ones, at frequencies relatively close to but separated from that of the initial resonator. Our aim was to provide estimates of the noise floor along with the measurement of the actual signal component. Figure 4 illustrates how the triplet resonator positions are set with respect to the initial ones, while figure 5 shows how adding the triplets changes the transfer function of a channel. Note that if the frequencies are too close to each other so the bandwidths of adjacent resonator channels overlap, this approach may cause resonances.
Figure 4. Visualization of the layout of the resonator triplets through the $z_i$ coefficients, see (2.5). $z_i = 1$ and $z_i = -1$ correspond to $f = 0$ and $f = f_s$, respectively. $f_1 = 30$ kHz, $f_s = 500$ kHz. The frequency difference between the initial resonators and the corresponding triplets was set to 5 kHz for clarity.

Figure 5. Comparison of the transfer functions of a single resonator channel in the normal and triplet resonator configurations. $\alpha = 0.001$. Notice the zeros in the transfer function on either side of the main peak for the triplet configuration, caused by the additional resonators.

4.5 Practical considerations for the frequency adaptation

Since the source of the parasitic signal we are aiming to detect is common for all eight channels processed by the same FPGA, we decided to implement frequency adaptation on one channel only and use the same $g_n$ and $c_n$ coefficients for all other channels as well.

For practical use, (3.1) needs some more elaboration. Technically, the phase function also crosses zero when it wraps from $\pi$ to $-\pi$ (or vice versa). These events, henceforth referred to as $\pi$ crossings, must not be considered as zero crossings. Excluding them, however, is very straightforward when using the algebraic form: we only register an event if $\text{Im} \{\hat{x}_{1,n}\}$ changes sign while $\text{Re} \{\hat{x}_{1,n}\} > 0$. In addition, the formula only gives the absolute value of the frequency difference. To obtain its sign, one also needs to consider the sign of the zero crossings registered: a positive zero crossing in phase is when $\text{Im} \{\hat{x}_{1,n}\}$ changes from negative to positive, which corresponds to $f_0 > f_r$, and the opposite for negative zero crossings.
Figure 6. Repetitive zero crossings due to noise in the phase of $\hat{x}_{1,n}$ calculated for an actual signal acquisition. $\circ$ corresponds to the single accepted positive zero crossing at the beginning, while the subsequent rejected positive and negative zero crossings are represented by $\times$ and $+$, respectively.

Noise in the input signal may cause spurious zero crossing events as demonstrated in figure 6. These events can be filtered out by requiring a $\pi$ crossing, that is, a sign change of $\text{Im}\{\hat{x}_{1,n}\}$ while $\text{Re}\{\hat{x}_{1,n}\} < 0$, to occur between any two accepted zero crossings.

All these considerations assume that the resonator value used for frequency adaptation rotates sufficiently slowly for the adequate detection of zero and $\pi$ crossings. A necessary condition for this is that the vector under consideration should never make a transition between non-adjacent quadrants of the complex plane. This requires the frequency mismatch to be inferior to $\frac{\pi}{4}$, a condition automatically fulfilled in most cases. Since we are tracking a slowly varying frequency with a relatively good initial estimate, we don’t expect failure cases related to this condition to occur.

The main motivation behind the development of the zero crossing-based frequency adaptation method was to avoid the division required for calculating the phase of the resonators, thus spare the FPGA resources that would be necessary for implementing a divider. However, according to (3.1), this method also requires a division at every adaptation step for calculating the estimator of the frequency difference. Nevertheless, by adapting the frequency whenever there is a suitable pair of zero crossings, the numerator of the fraction will always be 1. In this case, the division can be avoided by constructing a lookup table: for different values of $t_m$, storing the corresponding increment of the phase step size $\Delta \varphi$. The precision of these values could be improved by interpolation between the points of the lookup table if required. When adapting higher order harmonics, the resulting phase estimator values will need to be multiplied by the order of the harmonic. If desired, the phase estimators thus obtained can be averaged over several zero crossings.

4.6 Test results

The algorithm with frequency adaptation was tested on a development system operating in the laboratory and a prototype system installed at the Proton Synchrotron Booster (PSB) accelerator.

The frequency of the HVPS contributions varies with time. The main influence on the frequency appears to be temperature, as shown in figure 7. The data were acquired in the summer, when the temperature inside the building and, consequently, inside the rack, tends to follow the exterior
temperature. We see a corresponding cooldown overnight, echoed in the evolution of the HVPS frequency.

Figure 8 demonstrates the efficiency of the frequency adaptation in the laboratory system. The adaptation process is capable of following the slow drift of the frequency satisfactorily.

In our implementation, we use $\frac{\alpha}{N} = \frac{1}{50000}$, yielding a resonator bandwidth of approximately 5 Hz, which seems to be a good compromise between reduced noise in the acquisition and sufficient bandwidth for signal detection. Figure 9 shows the typical output signal of a resonator properly tuned to the frequency of the HVPS signal. The input signal of the observer is also shown for comparison. The peaks in the signal correspond to actual beam loss events. These events have
a clear impact on the output of the resonator, however, it recovers quickly and the influence of these peaks on the measurement can be eliminated by introducing a blind period after detecting a high peak. The resulting effect on frequency adaptation has been negligible in our acquisitions. In general, the level detected by the resonator properly tuned to the HVPS peak in the presence of HVPS contributions tends to be at least 20 dB higher than the signal in the other two resonators of the triplet, which provides a satisfactory margin for detection.

4.7 Merits of the zero crossing frequency adaptation scheme

We implemented the processing described above on an Altera Cyclone IV FPGA device present in the new BLM system. The system clock frequency was 100 MHz.

When using a fully serialized realization in order to minimize resource use, we implemented a functional observer with zero crossing based frequency adaptation using about 1300 logic cells and 21 9-kilobit memory blocks out of the device total of 149760 and 720, respectively. We estimated the resource use of a corresponding “original” adaptive Fourier analyzer (AFA) implementation including a divider at about 2450–2800 logic cells, with 12 9-kilobit memory blocks. At the price of increased memory use, the zero crossing scheme promises a saving in logic cells on the order of 50%.

At our system clock frequency, the typical latency of the divider required for the calculations of the “original” AFA is about 15 clock cycles. If a realization computing all resonators in parallel were implemented, depending on the number of resonators, our observer could process one new input sample in about 15–20 clock cycles. Part of the latency of the divider could be spared, but the “original” AFA would still take approximately 25–30 clock cycles to process a sample. Thus, with our scheme based on zero crossings, we can expect a gain of up to 50% in terms of maximum input sample rate, at the price of increased FPGA hardware use.

Since the estimator of the frequency is only updated at zero crossings of the resonator phase, it may be less susceptible to apparent frequency changes resulting from the influence of noise in the signal. Figure 10 shows an example with a noisy input signal where the zero crossing adaptation...
Figure 10. Example: frequency estimator yielded by the zero crossing-based frequency adaptation method in comparison to some other adaptation schemes with a noisy input signal. The input signal is a sinusoid with linearly varying frequency from 29.95 kHz to 30.05 kHz contaminated with white Gaussian noise, SNR = 20 dB. In the resonator structure, $\alpha = 1$, $f_1 = 30$ kHz, $f_s = 500$ kHz, thus $N = 17$. In this configuration, the fluctuations exhibited by the frequency estimator obtained with the “original” AFA are about an order of magnitude higher than with the block AFA, therefore the former curve is omitted for clarity. For the robust AFA, we used a damping factor of $\eta = 32$. The block AFA was executed with a measurement period $P = 333$. The scheme yields a considerably smoother frequency estimator than some of the other frequency adaptation methods.

5 Conclusions

We presented a principle to complement an existing Fourier analyzer implementation with a resource-efficient frequency adaptation method, practical considerations for realizing it and a use case with operational experience. The method promises implementational advantages or reduced noise in the frequency estimator at the price of a less reactive frequency adaptation. It can be implemented efficiently in applications where the computational complexity is critical, especially on synthesized hardware. The performance delivered by the scheme appears suitable for the application we tested it with.

Acknowledgments

Csaba F. Hajdu would like to express his gratitude to William Viganò for his continual help with both the theoretical and practical aspects of this project.

Tamás Dabóczi acknowledges the support of ARTEMIS JU and the Hungarian National Research, Development and Innovation Fund in the frame of the R5-COP project.

References


