Memory-efficient high-speed algorithm for multi-τ PDEV analysis

Magnus Danielson
R&D System Design
Net Insight AB
Stockholm, Sweden
Email: magda@netinsight.net

François Vernotte
Observatory THETA/UTINAM,
UBFC/UFC and CNRS
Besançon, France
Email: francois.vernotte@obs-besancon.fr

Enrico Rubiola
CNRS FEMTO-ST Institute,
Dept Time and Frequency
Besançon, France
Email: rubiola@femto-st.fr

***Abstract*—The Ω preprocessing was introduced to improve phase noise rejection by using a least square algorithm. The associated variance is the PVAR which is more efficient than MVAR to separate the different noise types. However, unlike AVAR and MVAR, the decimation of PVAR estimates for multi-τ analysis is not possible if each counter measurement is a single scalar. This paper gives a decimation rule based on two scalars, the processing blocks, for each measurement. For the Ω preprocessing, this implies the definition of an output standard as well as hardware requirements for performing high-speed computations of the blocks.**

***Index Terms*—Least square methods, Phase noise, Stability analysis, Time-domain analysis.**

I. PVAR AND Ω-COUNTERS

The concept of Ω-counter was formulated by Rubiola [1], based on Johansson [2], to achieve the optimal rejection of white phase noise for short term frequency measurement by using an estimator based on the least squares. Such methods was presented by Barnes [3], for the purpose of drift estimation under presence of white noise. The algorithms of [4] does not provide means of decimation of data while maintaining the least square properties, it rather states that there is no such method known. In [5] such method was presented, thus allowing for decimation.

The principle of this frequency estimation is to calculate the least squares slope over a phase sequence $\{x_k\}$ obtained at instants $t_k = k\tau_0$ with $k \in \{0, \ldots, N-1\}$ where τ_0 is the sampling step and $\tau = (N-1)\tau_0$ the total length of the sequence. It is well known that the least squares provide the best slope estimate in the presence of white noise (i.e. white PM noise) [3]. It has been demonstrated that, in the presence of white PM, the variance of this frequency estimate is lower by a factor of $\frac{3}{4}$ than the MVAR. Moreover, since the least squares are optimal for white noise, the variance of the Ω-counter estimate is minimal. It is then an efficient estimator [6].

The Ω-counter weight functions for phase data as well as for frequency deviations are plotted in Figure 1. The shape of $w_c(t)$ (see Figure 1-B) explains the choice of the Greek letter Ω to name this counter [7], [1].

PVAR is then defined as $\text{PVAR}(\tau) = \frac{1}{2}\left\langle \left(\hat{\mathbf{y}}_2^\Omega - \hat{\mathbf{y}}_1^\Omega\right)^2 \right\rangle$ [4]. The weight function associated to PVAR for phase data is plotted in Figure 2.

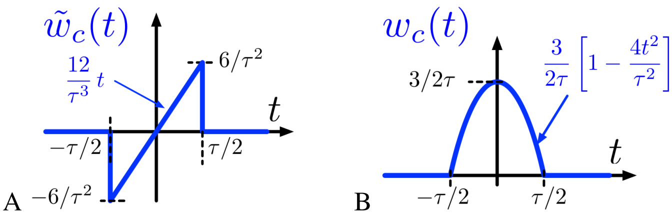

Fig. 1. weight functions of the Ω-counter computed from phase data (A, left) or from frequency deviations (B, right).

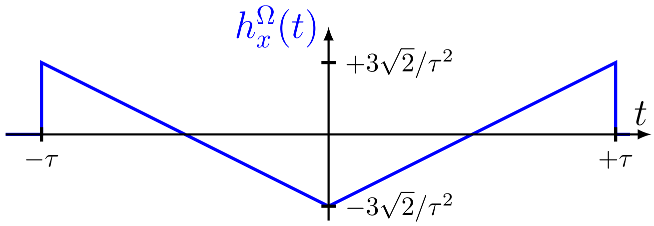

Fig. 2. weight function associated to PVAR for phase data.

PVAR, like MVAR, is intended to deal with short term analysis (and then white and flicker PM noises) whereas AVAR is preferred for the measurement of long term stability and timekeeping. The main advantage of PVAR regarding MVAR relies on the larger EDF of its estimates, and in turn the smaller confidence interval. The best of PVAR is its power to detect and identify weak noise processes with the shortest data record. PVAR is superior to MVAR in all cases, and also superior to AVAR for all short-term and medium-term processes, up to flicker FM included. AVAR is just a little better with random walk and drift. Therefore, PVAR should be an improved replacement for MVAR in all cases, provided the computing overhead can be accepted.

Thus, the only drawback of PVAR lies in the difficulty to find its decimation algorithm. In order to solve this problem, let us remind the basics of decimation.

II. LEAST-SQUARE FREQUENCY ESTIMATION

A. Linear system

The least square system producing the output vector $\mathbf{x}$ of phase samples from the system state vector $\mathbf{c}$ using the system matrix $\mathbf{A}$ and assuming the error contribution of $\mathbf{d}$ as defined in $\mathbf{x} = \mathbf{A}\mathbf{c} + \mathbf{d}$ having the least square estimation as given by

$$\hat{\mathbf{c}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{x} \tag{1}$$

For this system, a linear model of phase and frequency state is defined

$$\hat{\mathbf{c}} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \tag{2}$$

A block of phase samples, taken with τ_0 time in-between them, building the series x_n where n is in the range $\{0, \ldots, N-1\}$ where by convention N is the number of phase samples. In the system model, each sample n has an associated observation time $t_n = \tau_0 n$. The matrix $\mathbf{A}$ and the vector $\mathbf{x}$ then becomes

$$\mathbf{A} = \begin{pmatrix} 1 & t_0 \\ \vdots & \vdots \\ 1 & t_n \\ \vdots & \vdots \\ 1 & t_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \vdots & \vdots \\ 1 & \tau_0 n \\ \vdots & \vdots \\ 1 & \tau_0(N-1) \end{pmatrix} \tag{3}$$

$$\mathbf{x} = \begin{pmatrix} x_0 \\ \vdots \\ x_n \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{4}$$

B. Closed form solution

Inserting (3) and (4) into (1) results in

$$\hat{\mathbf{c}} = \left[\begin{pmatrix} \cdots & 1 & \cdots \\ \cdots & \tau_0 n & \cdots \end{pmatrix} \begin{pmatrix} \vdots & \vdots \\ 1 & \tau_0 n \\ \vdots & \vdots \end{pmatrix} \right]^{-1} \times \begin{pmatrix} \cdots & 1 & \cdots \\ \cdots & \tau_0 n & \cdots \end{pmatrix} \begin{pmatrix} \vdots \\ x_n \\ \vdots \end{pmatrix} \tag{5}$$

simplifies into

$$\hat{\mathbf{c}} = \begin{pmatrix} \sum_{n=0}^{N-1} 1 & \tau_0 \sum_{n=0}^{N-1} n \\ \tau_0 \sum_{n=0}^{N-1} n & \tau_0^2 \sum_{n=0}^{N-1} n^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum_{n=0}^{N-1} x_n \\ \tau_0 \sum_{n=0}^{N-1} n x_n \end{pmatrix} \tag{6}$$

replacing the sums C and D

$$C = \sum_{n=0}^{N-1} x_n \tag{7}$$

$$D = \sum_{n=0}^{N-1} n x_n \tag{8}$$

becoming

$$\hat{\mathbf{c}} = \begin{pmatrix} N & \tau_0 \frac{N(N-1)}{2} \\ \tau_0 \frac{N(N-1)}{2} & \tau_0^2 \frac{N(N-1)(2N-1)}{6} \end{pmatrix}^{-1} \begin{pmatrix} C \\ \tau_0 D \end{pmatrix} \tag{9}$$

inverse can be solved as

$$\begin{pmatrix} N & \tau_0 \frac{N(N-1)}{2} \\ \tau_0 \frac{N(N-1)}{2} & \tau_0^2 \frac{N(N-1)(2N-1)}{6} \end{pmatrix}^{-1} = \frac{12}{\tau_0^2 N(N-1)(N+1)} \times \begin{pmatrix} \tau_0^2 \frac{(N-1)(2N-1)}{6} & -\tau_0 \frac{N-1}{2} \\ -\tau_0 \frac{N-1}{2} & 1 \end{pmatrix} \tag{10}$$

insertion of (2) and (10) into (9) resulting in the estimators

$$\hat{x} = \frac{6}{N(N+1)} \left(\frac{(2N-1)}{3} C - D \right) \tag{11}$$

$$\hat{y} = \frac{12}{\tau_0 N(N-1)(N+1)} \left(-\frac{N-1}{2} C + D \right) \tag{12}$$

these estimators have been verified to be bias free from static phase and static frequency, as expected from theory. Using these estimator formulas the phase and frequency can estimated of any block of N samples for which the C and D sums have been calculated.

C. PVAR calculation

The PVAR estimator calculation is defined from the equations

$$\hat{\sigma}_P^2(\tau) = \frac{1}{M} \sum_{i=1}^{M} (\alpha_i)^2 \tag{13}$$

$$\alpha_i = \frac{1}{\sqrt{2}} \left(\hat{\mathbf{y}}_i^{\Omega} - \hat{\mathbf{y}}_{i+1}^{\Omega} \right) \tag{14}$$

inserting (12) and (14) into (13) produces

$$\hat{\sigma}_P^2(\tau) = \frac{72}{M \tau_0^2 N^2 (N-1)^2 (N+1)^2} \sum_{i=1}^{M} \left[(D_i - D_{i+1}) - \frac{N-1}{2} (C_i - C_{i+1}) \right]^2 \tag{15}$$

where (C_i, D_i) and (C_{i+1}, D_{i+1}) is two pairs of accumulated sums being consecutive. These may be either forms by the direct accumulation of (7) and (8) or through the decimation rule of (25) and (26), as long as N is the number of samples in each block (being of equal length) and that the block observation time $\tau = N\tau_0$. Using the decimation rules, any τ calculation can be produced and then their PVAR calculated using (15). Notice that M is the number of averaged blocks.

III. Decimation

A. Decimation of different sized blocks

The key idea in decimation is to form the (C, D) pair for a larger set of samples. Consider a block of N_{12} samples. The definition says

$$C_{12} = \sum_{n=0}^{N_{12}-1} x_n \tag{16}$$

$$D_{12} = \sum_{n=0}^{N_{12}-1} n x_n \tag{17}$$

but for practical reason processing is done on two sub blocks being N_1 and then N_2 samples long, giving

$$N_{12} = N_1 + N_2 \tag{18}$$

$$C_1 = \sum_{n=0}^{N_1-1} x_n \tag{19}$$

$$C_2 = \sum_{n=0}^{N_2-1} x_{N_1+n} \tag{20}$$

$$D_1 = \sum_{n=0}^{N_1-1} n x_n \tag{21}$$

$$D_2 = \sum_{n=0}^{N_2-1} n x_{N_1+n} \tag{22}$$

the C_{12} sum can be reformulated as

$$\begin{aligned} C_{12} &= \sum_{n=0}^{N_{12}-1} x_n \\ &= \sum_{n=0}^{N_1-1} x_n + \sum_{n=N_1}^{N_{12}-1} x_n \\ &= C_1 + \sum_{n=0}^{N_2-1} x_{N_1+n} \\ &= C_1 + C_2 \end{aligned} \tag{23}$$

where N_1 can be chosen arbitrarily under the assumption $0 \leq N_1 \leq N_{12}$ and then $N_2 = N_{12} - N_1$. Similarly the D_{12} sum can be reformulated as

$$\begin{aligned} D_{12} &= \sum_{n=0}^{N_{12}-1} n x_n \\ &= \sum_{n=0}^{N_1-1} n x_n + \sum_{n=N_1}^{N_{12}-1} n x_n \\ &= D_1 + \sum_{n=0}^{N_2-1} (N_1 + n) x_{N_1+n} \\ &= D_1 + \sum_{n=0}^{N_2-1} N_1 x_{N_1+n} + \sum_{n=0}^{N_2-1} n x_{N_1+n} \\ &= D_1 + N_1 \sum_{n=0}^{N_2-1} x_{N_1+n} + D_2 \\ &= D_1 + N_1 C_2 + D_2 \end{aligned} \tag{24}$$

Thus using (23) and (24) any set of consecutive blocks can be further decimated to form a new longer block. For each decimation, only the length N and sums C and D needs to be stored, thus reducing the memory requirements. In a pre-processing stage, these sums can be produced. The decimation rule thus allows for any length being a multiple to the pre-processed length to be produced, with maintained non-biased phase, frequency and PVAR estimator properties.

B. Decimation by N

The generalized decimate by N formulation follows natural from this realization and is proved directly though recursively use of the above rule. Consider that a preprocessing provides C and D values for block of length N_{pre}, then on first decimation block 0 and 1 is decimated, and block 1 needs to be raised with $N_{\text{pre}} C_1$ (as illustrated in Figure 4), as block 2 is decimated in the next round, $2N_{\text{pre}} C_2$ etc, and in general we find

$$C_{tot} = \sum_{i=0}^{N_2-1} C_i \tag{25}$$

$$D_{tot} = \sum_{i=0}^{N_2-1} D_i + i N_1 C_i \tag{26}$$

for the observation time $\tau = \tau_0 N_1 N_2$ with $N_1 N_2$ samples, for use with the (11) and (12) estimators.

This decimation by N mechanism can be used together with the generic block decimation to form any form of block processing suitable, thus providing a high degree of freedom in how large amounts of data is being decimated.

C. Geometric representation

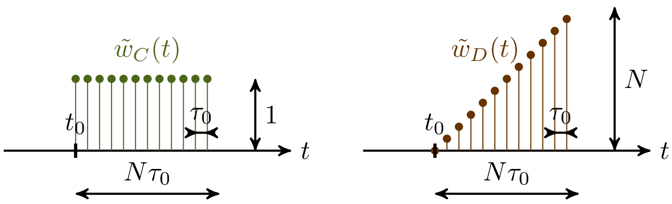

Fig. 3. weight functions of the C and D elementary block pair.

1) Decimation rule: Figure 3 represents the weight functions of the C and D elementary block pair that we will symbolize respectively with and .

In the same way as in § III-A, let us consider two consecutive sets of N_1 samples, beginning respectively at instants t_1 and t_2, and the whole sequence of $N_{12} = 2N_1$ samples. We can form the blocks C_1 and D_1 over the first sub-sequence, C_2 and D_2 over the second one as well as C_{12} and D_{12} over the whole sequence (see left hand side of Figure 4). The right hand side of Figure 4 shows that $C_{12} = C_1 + C_2$ and $D_{12} = D_1 + N_1 C_2 + D_2$ as demonstrated in (23) and (24).

2) The Ω-counter weight function: As stated in § I, the weight function of the Ω-counter for phase data $x(t)$ is given by [4]. Figure 5 shows that the estimate $\hat{\mathbf{y}}^\Omega$ of the Ω-counter is:

$$\hat{\mathbf{y}}^\Omega \propto D - \frac{N}{2} C.$$

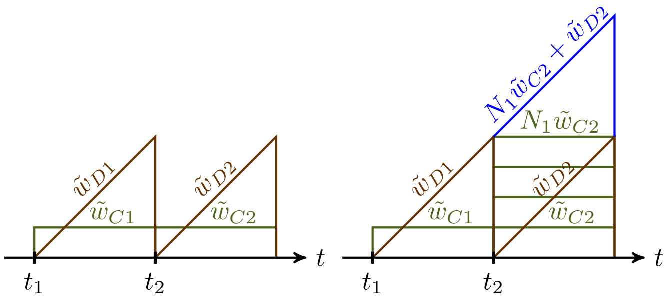

Fig. 4. Decimation rule of the (C_1, D_1) and (C_2, D_2) block pair weight functions over two adjacent sub-sequences for composing the (C_{12}, D_{12}) block pair weight functions over the whole sequence.

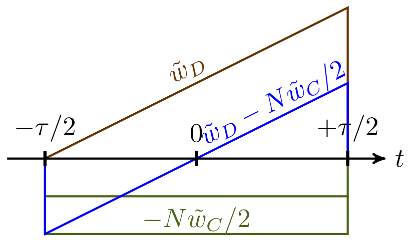

Fig. 5. Association of the (C, D) block pair weight functions for composing the Ω-counter weight function.

3) The PVAR weight function: Since $\text{PVAR}(\tau) = \frac{1}{2}\left\langle\left(\hat{\mathbf{y}}_2^{\Omega} - \hat{\mathbf{y}}_1^{\Omega}\right)^2\right\rangle$, it comes

$$\text{PVAR}(\tau) \propto \left\langle\left(D_2 - \frac{N}{2}C_2 - D_1 + \frac{N}{2}C_1\right)^2\right\rangle$$

as illustrated by Figure 6.

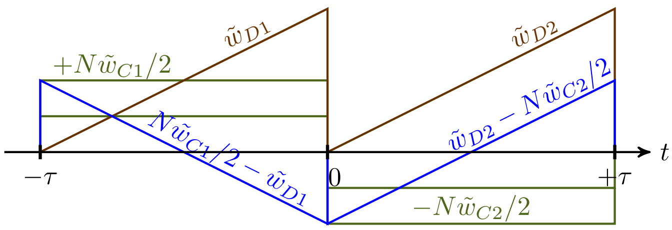

Fig. 6. Association of the (C_1, D_1) and (C_2, D_2) block pair weight functions for composing the PVAR weight function.

D. Decimation processing

It should be noted that the decimation process may be used recursively, such that it is used as high-speed preprocessing in FPGA and that the (C, D) pairs is produced for each N_1 samples as suitable for the plotted lowest τ. Another benefit of the decimation processing is that if the FPGA front-end has a limit to the number of supported N_1 it can process, software can then continue the decimation without causing a bias. This provides for a high degree of flexibility without suffering from high memory requirements, high processing needs or for that matter overly complex HW support.

E. Multi-τ decimation

Another aspect of the decimation processing is not only that many τ can be produced out of the same sample or block sequence, but once a suitable set of τ variants have been produced, these can be decimated recursively in suitable form to create τ variants of higher multiples. One such approach would be to produce the 1 to 9 τ multiples (or only 1, 2 and 5 multiples which is enough for a log-log plot) of accumulates for τ being 1 s, thus producing the 1 to 9 s sums, and by recursive decimation by 10 produces the same set of points on the log-log plot, but for 10^{th} multiple of time, for each recursive step. This will allow for large ranges of τ to be calculated for a reasonable amount of memory and calculation power.

IV. Summary

Presented is an improved method to perform least-square phase, frequency and PVAR estimates, allowing for high speed accumulation similar to [8], but extending into any τ needed. It also provides for multi-τ analysis from the same basic accumulation. The decimation method can be applied recursively to form longer τ estimates, reusing existing calculations and thus saving processing. Thus, it provides a practical method to provide PDEV log-log plots, providing means to save memory and processing power without the risk of introducing biases in estimates, as previous methods have shown.

References

[1] E. Rubiola, M. Lenczner, P.-Y. Bourgeois, and F. Vernotte, "The Omega counter, a frequency counter based on the linear regression," *IEEE UFFC*, 2016, submitted (see arXiv:1506.05009).

[2] S. Johansson, "New frequency counting principle improves resolution," *Proceedings of 37th PTTI*, 2005, http://spectracom.com/sites/default/files/document-files/Continuous-timestamping-article.pdf.

[3] J. Barnes, "The measurement of linear frequency drift in oscillators," *Proceedings of 15th PTTI*, 1983, http://tycho.usno.navy.mil/ptti/1983papers/Vol 15_29.pdf.

[4] F. Vernotte, M. Lenczner, P.-Y. Bourgeois, and E. Rubiola, "The parabolic variance (PVAR), a wavelet variance based on the least-square fit," *IEEE UFFC*, 2016, accepted (see arXiv:1506.00687).

[5] M. Danielson, F. Vernotte, and E. Rubiola, "Least square estimation of phase, frequency and pdev," 2016, (see arXiv:1604.01004).

[6] B. S. Everitt, "The cambridge dictionary of statistics," *Cambridge University Press*, 1998.

[7] E. Rubiola, "On the measurement of frequency and of its sample variance with high-resolutioin counters," *RSI, vol. 76, no. 5*, May 2005, also arXiv:physics/0411227, Dec. 2004.

[8] J. J. Snyder, "An ultra-high resolution frequency meter," *Proceeedings of 35th Annual Frequency Control Symposium*, May, 1981.