P-Class Phasor Measurement Unit Algorithms Using Adaptive Filtering to Enhance Accuracy at Off-Nominal Frequencies

Andrew J. Roscoe, Ibrahim F. Abdulhadi,
Graeme M. Burt
Department of Electronic and Electrical Engineering

University of Strathclyde
Glasgow, UK
Andrew.Roscoe@eee.strath.ac.uk

Abstract— **While the present standard C.37.118-2005 for Phasor Measurement Units (PMUs) requires testing only at steady-state conditions, proposed new versions of the standard require much more stringent testing, involving frequency ramps and off-nominal frequency testing. This paper presents two new algorithms for "P Class" PMUs which enable performance at off-nominal frequencies to be retained at levels comparable to the performance for nominal frequency input. The performances of the algorithms are compared to the "Basic" Synchrophasor Estimation Model described in the new standard. The proposed algorithms show a much better performance than the "Basic" algorithm, particularly in the measurements of frequency and rate-of-change-of-frequency at off-nominal frequencies and in the presence of unbalance and harmonics.**

Keywords-component; Power system measurements; Discrete Fourier transforms; Filtering algorithms; Power system harmonics

I. INTRODUCTION

Closed-loop real-time applications of synchrophasor measurements are gaining momentum. Examples of these include system integrity protection schemes (SIPS) and control of FACTS (Flexible AC transmission system) devices to improve system stability margins [1, 2]. Achieving such functionality, however, requires accurate transient information as reported by the PMUs (Phasor Measurement Units). Wide-area protection algorithms such as those based on ROCOF (Rate of change of frequency) to perform selective load shedding also rely on accurate, timely measurements, potentially in an environment of localised frequency deviations in the aftermath of a major disturbance.

Validation of power system models using synchrophasors also necessitates an accurate transient response since this information can be used for tuning wide-area generator and FACTS controllers. Furthermore, disparity in dynamic behaviour of PMUs from different vendors presents a challenge to achieving interoperability in digital substations. Should the protection and control schemes rely on synchrophasors from sources of dissimilar fidelity, mal-operation may occur.

The existing standard for PMUs is given by C.37.118-2005 [3]. This standard does not require testing under any non-stationary conditions [4]. The proposed new standard is under development and currently stands at PC37.118.1(Draft 3.0) [5]. This standard now requires testing under frequency ramps to ±1 Hz/s over a ±2 Hz window centered on the nominal frequency, amplitude steps, and phase steps. The algorithms are expected to be compliant with measurement accuracy specifications when 1% of any harmonic up the 50th are present (for P class) or 10% of any such harmonic is present (for M class). The relationship between the measurements and the timestamps have also been specified much more formally, as have the latencies of the filters. Lack of clarity in the previous standard allowed PMUs from different manufacturers to behave in quite different manners but to still be compliant [6]. The new standard places much stricter requirements on the design of the PMU algorithm and its calibration mechanisms.

The new standard describes "Basic" reference algorithms for P-class (protection) and M-class (measurement) devices, which are claimed to meet the requirements. This paper explores the behaviour of the "Basic" P class PMU algorithm, and also describes two alternative algorithms which provide much better performance for off-nominal frequencies, particularly where the measurement of frequency and rate-of-change-of-frequency (ROCOF) are concerned. This is achieved by substituting the fixed-weight FIR (Finite Impulse Response) filter in the "Basic" algorithm with cascaded adaptive comb filters, created using rectangular exact-time average sections. These are fast to execute in real time and are shown to be viable at up to (or beyond) the 10 kHz sample rate.

The use of adaptive filtering introduces additional effects and feedback paths within the PMU algorithm. These are described below, along with the algorithmic adjustments necessary to make the algorithms robust and accurate.

II. THE BASIC MODEL FROM PC37.118.1 (DRAFT 3.0)

The "Basic" model for the P-Class PMU described in the emerging PC37.118.1 standard is shown in Figure 1 and Figure 2. In the "Basic" architecture, the input signals are correlated with a waveform at the nominal frequency f_0. The output of each single-phase section is a phasor, each of which has a magnitude proportional to the voltage on each phase. The phase angles of each phasor rotate at a rate of $2\pi(f-f_0)$ where f_0 is the correlation frequency (the nominal system frequency), and f is the actual system frequency.

The work has been carried out as part of the Rolls-Royce UTC "programme", and has also received funding from the European Union Framework 7 program on the basis of Decision No 912/2009/EC.

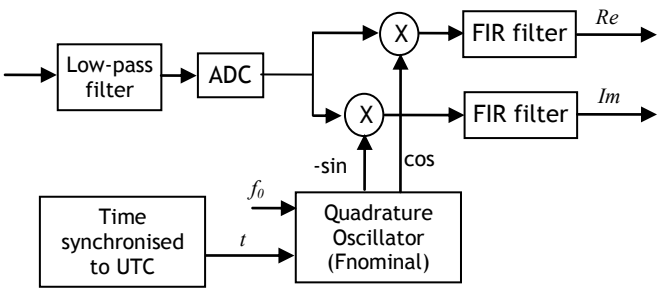

Figure 1. Single-phase section of the "Basic" P-Class PMU

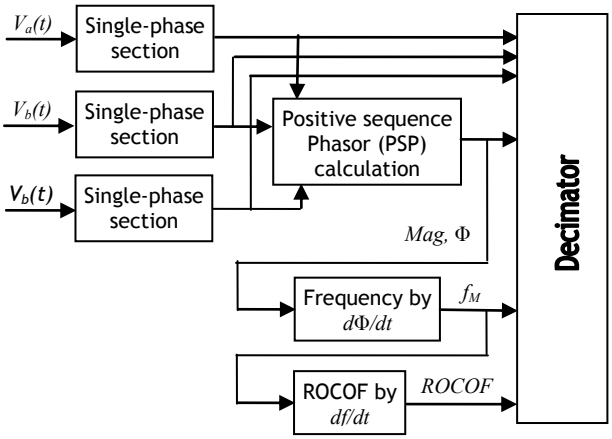

Figure 2. Three-phase "Basic" PMU

During normal balanced operation, the phase angles of the Va, Vb, and Vc phasors are at (or close to) 120° to each other. The overall positive-sequence phasor can be calculated by:

$$\mathbf{V^p} = \mathbf{V_a} + \mathbf{V_b} e^{\frac{2\pi}{3}j} + \mathbf{V_c} e^{-\frac{2\pi}{3}j} \qquad \textbf{(1)}$$

Assuming that the single-phase sections are effective at filtering out noise, harmonics, and other unwanted non-harmonic signals from $\mathbf{V_a}$, $\mathbf{V_b}$ and $\mathbf{V_c}$, then the phasor $\mathbf{V^p}$ will also rotate at a steady rate of $2\pi(f-f_0)$, for steady state values of f, and if the input signal magnitudes and relative phases remain constant.

In the "Basic" P-class model, the FIR (Finite Impulse Response) filter used is a fixed-length triangular-weighted symmetric filter of length 2 cycles, designed to work optimally at the nominal system frequency f_0. The filter produces notches with high attenuation at every multiple of f_0, which are useful to attenuate contamination due to harmonics. For example, the filter weights are determined by:

$$W_k = \left(1 - \frac{2}{(N+2)}|k|\right) \qquad \textbf{(2)}$$

where:

$$k = -\frac{N}{2}, -\frac{N}{2}+1, \ldots . \frac{N}{2}$$

N = filter order, where $N = 2(S-1)$ and S is the number of samples per cycle at nominal frequency f_0.

An example is given in [5] for a filter with 15 samples per cycle, giving an order 28 FIR filter. This is reproduced below:

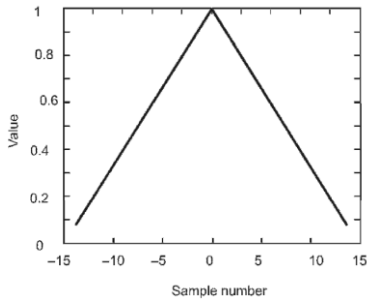

Figure 3. P class FIR filter coefficient weights (29 off) for S=15 samples/cycle and order N=28 [5]

Since the filter is symmetric, its response is "zero phase" to an input waveform with a steady frequency, if the centre of the filter is placed at a time reference of zero. Of course, in practice, any real filter must be causal and only the ADC samples from the present instant and those in the past are known. However, the "zero phase" property can still be used because the standard allows the timestamp of the measurement to be allocated as if the measurement actually occurred half-way through the FIR filter time length, where "time zero" would be placed for a non-causal filter.

There are 2 problems with such an implementation, both of which are identified in the emerging standard and also in [7]. The problems become evident when the mixing frequency is considered. The mixing frequency appears at the inputs to the FIR filters in Figure 1. If frequency is nominal, i.e. $f=f_0$, then the mixing frequency consists of the dominant (wanted) DC component, plus unwanted components at $f+f_0=2f_0$:, and at every frequency $f_H=f_0\pm Hf_0$ for the harmonics where H>1 and $H \in \mathrm{N}$. However, when the frequency is not nominal, then the wanted component is no longer at DC, but is present at $|f-f_0|$ Hz. The unwanted harmonic components also shift from $f_H=f_0\pm Hf_0$ *to* $f_H=f_0\pm Hf$.

1) For off-nominal frequencies, the FIR filter notches no longer correspond exactly to the unwanted frequencies in the mixed signal. Therefore, the ability of the FIR filter to reject harmonic contamination reduces as frequency diverges from nominal.
2) While the "Basic" FIR filter is carefully designed to be symmetric and "zero phase", it has a finite amplitude attenuation of the wanted component when $f \neq f_0$, i.e. the mixing frequency is not 0 (DC) but is finite. Therefore, the measured amplitude needs to be calibrated.

It would, in theory, be possible to address 1) by carefully designing new filters (in real time) to place notches at the

desired frequencies using, for example, the Tustin transformation [8] or other mathematical methods. However, designing the FIR in this manner is likely to be a time-consuming process. Also, it (alone) does not address 2).

This paper instead proposes two different P-class PMU algorithms which address both 1) and 2) by allowing the correlation frequency to adjust in real-time, and by implementing adaptive filters based upon cascaded exact-time averaging algorithms [9-11]. These can be executed extremely quickly in real-time since the FIR filter weights and zero/pole placements do not need to be recalculated explicitly. The proposed method of cascaded exact-time average filters also executes much faster than a traditional FIR filter, since only the effects of the first and last samples have to be calculated for each stage at each computational frame. This contrasts with a traditional FIR filter implementation in which the entire FIR filter window correlation needs be calculated at each computational frame.

III. TWO PROPOSED P-CLASS PMU ALGORITHMS

A. *Symmetric twin filters in a "tick-tock" algorithm*

The first proposed algorithm maintains a symmetric FIR filter shape. This maintains the "zero-phase" filter property, if the centre of the filter window is considered to be at the timestamp. However, the correlation frequency f_C is allowed to move with the measured frequency instead of remaining fixed at f_0 , so that the wanted mixing frequency remains at DC or extremely close to it. The unwanted harmonic components then fall at harmonics of $f_H = f \pm Hf$. Compared to the "Basic" algorithm, this is useful because now the required FIR filter notch frequencies to attenuate harmonics and unbalanced effects are at exact integer multiples of the fundamental frequency. This makes the notch filter much easier to implement.

To achieve a symmetrical Fourier correlation and FIR filter when the frequency is changing requires a pair of correlation and FIR filter paths to operate in parallel (Figure 4). The correlation frequency f_C and the filter in one path needs to be configured with the measured estimate of the system frequency f_M*, but this configuration then needs to be held constant so that the FIR filter behaviour is fixed and symmetric for a certain period of time. The filter needs to be allowed to settle for at least 2 cycle periods (the P-class filter length) so that the FIR filter output is fully consistent and symmetric. After this, the filter can be used. This "frees up" the other filter path, which is then reconfigured to the latest frequency measurement, and allowed to settle, etc. This creates a "tick-tock" pair of filter paths which are alternately used and reconfigured/settled to create a seamless output, but minimising deviation of the mixing frequency from 0 Hz.

Each FIR filter path itself (Figure 5) is constructed using 2 cascaded exact-time averaging filters, each of which averages its input waveform over exactly one cycle (a time period of $1/f_C$) at the correlation frequency f_C . This can be called a "1+1" filter. The averaging algorithms have been careful constructed to work accurately and quickly, being able to interpolate between samples where necessary [11]. Each (identical) single-cycle filter section has a rectangular shape, and causes a notch at every multiple of the correlation frequency. The zeros and poles are not explicitly calculated in real-time, but "automatically" fall into the appropriate places. For example, Figure 6 shows the pole-zero plot for such a single-cycle averaging filter, and also the amplitude responses for a single filter and for 2 filters cascaded. The overall filter shape in the time domain for the 2-filter cascade is the convolution of the 2 rectangular filters, and forms a triangular filter. When the measured frequency is nominal, i.e. $f = f_M* = f_C = f_0$, the cascaded filter is exactly equivalent to the triangular filter for the "Basic" PMU.

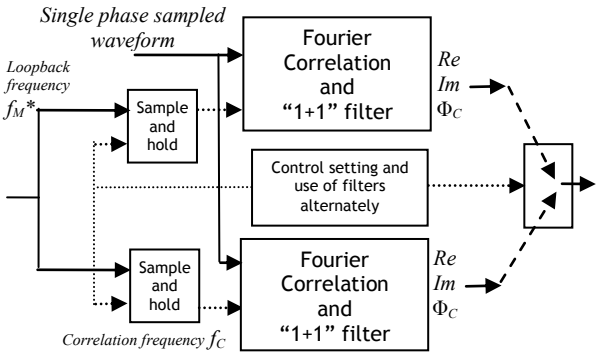

Figure 4. A pair of correlation and symmetric filter paths in a "tick-tock" arrangement

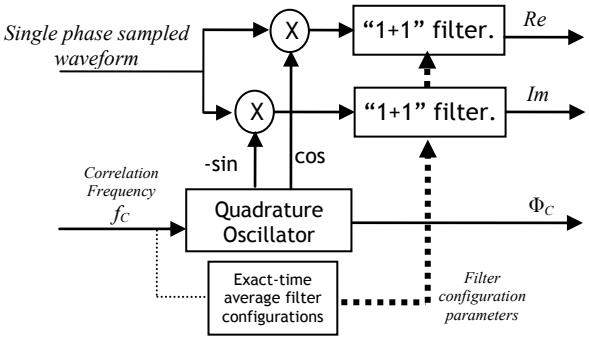

Figure 5. Single-phase correlation and "1+1" filter, adaptive

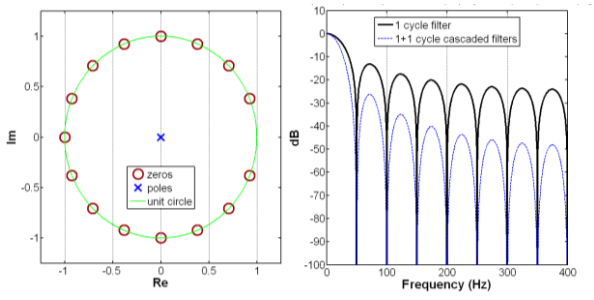

Figure 6. Example of zeros, poles and response for single-cycle averaging filter s (0.02s window length per filter, 800 Hz sampling)

To evaluate the phasor phase, relative to the reference phase at f_0, it needs to be appreciated that the phase of the real and imaginary components from the chosen "tick-tock" filter path represents the average phase of the signal, relative to Φ_C, where Φ_C is the phase of the correlation waveform. Φ_C accumulates over time at a rate of $2\pi f_C$.. Therefore, since the timestamp of the measurement is to be in the middle of the time window, careful back-tracking and calibration must be done:

$$t_{Timestamp} = t_{Now} - \frac{t_{FilterWindowLength}}{2} = t_{Now} - \frac{1}{f_C} \quad (3)$$

$$\Phi_{PhasorRelReference} = \operatorname{atan2}(Im, Re) + \left(\Phi_C - \frac{2\pi f_C}{f_C}\right) - 2\pi f_0 t_{Now} \quad (4)$$

Note that in (4), for the special case of the 2-cycle filter, Φ_C at the timestamp is exactly 2π (i.e. 0) less than Φ_C at the present time. It should also be noted that during rate-of-change of frequency (ROCOF) events, the frequency of the input signal appears as a chirp. Since the length of the P-class filter is only 2 cycles long, the effect on the measured phase can be ignored. For M-class devices however, this additional effect must be carefully accounted for, but this is beyond the scope of this paper.

The measured amplitude can also be corrected by estimating the (almost-zero) mixing frequency from the correlation , as it passes through the two rectangular time-window average filters, which introduce a cumulative gain of:

$$Gain = \left(\frac{\sin\left(\frac{\pi(f_M - f_c)}{f_c}\right)}{\frac{\pi(f_M - f_c)}{f_c}} \right)^2 \quad (5)$$

The measured amplitudes and phases also need to be corrected to account for analogue input hardware behaviour, such as instrumentation and anti-aliasing filters etc.

The positive sequence calculation, and deduction of measured frequency f_M and ROCOF by differentiating against time, are carried out in a similar fashion to the "Basic" algorithm. However, some care is needed at the instant that the "tick-tock" filter path change-over occurs. The high-level architecture is actually similar to Figure 7 (the Asymmetric algorithm, described in the next section) but the filter paths are doubly complex as per Figure 4. To measure frequency, the best performance is achieved by ignoring the phase calibrations for hardware, since this removes feedback paths for noise and interference. This requires two separate positive-sequence calculations: one using only partially calibrated phasors for the frequency calculation, and one using fully calibrated phasors for the outputs of single-phase and positive-sequence magnitude and phase.

The measured frequency f_M can be looped back (with a single cycle delay to avoid "algebraic loops"), and be used to calculate calibration factors and to determine filter settings. However an extra filter step to avoid oscillations and undesirable feedback is to further time-average the measured frequency f_M over a period of approximately 3 cycles before looping it back as f_M*. The length of this extra averaging is not critical, although making it exactly 3 cycles by averaging over exactly $3/f_M$* seconds adds further to the rejection of harmonics. The extra averaging does not impact on the latency of the primary PMU outputs, since the looped-back frequency f_M* is only used internally to configure the correlations and filters.

B. *Asymmetric filter algorithm*

While the symmetric filter algorithm produces a very good performance, it is memory-hungry and relatively complex, due to the requirement for 2 parallel sets of correlation and filter paths, and the mechanisms for switching seamlessly between the two paths.

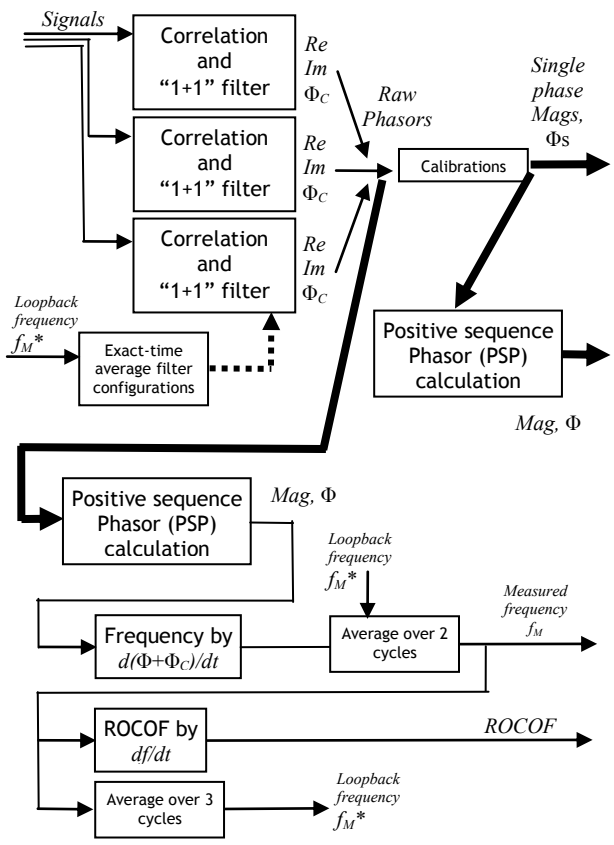

Figure 7. Asymmetric filter algorithm overview

An alternative approach is to allow the FIR filters to become asymmetric. This means that the "zero phase" property is lost. However, it turns out that this can be dealt with relatively easily. The entire algorithm logic is summarised in Figure 7, which shares some features with the "Frequency

locked loop" FLL algorithm previously developed in [12]. The structure is similar to the Symmetric filter algorithm, except that only one set of filters is required instead of the pair of "tick-tock" filters, and that the calibration process is slightly different.

There are 2 positive sequence calculation blocks in Figure 7. This is because there is a feedback path from the measured frequency via the loopback frequency f_M* to many filters within the algorithm. This loopback path creates an IIR (infinite impulse response) response in f_M* which must not be allowed to become unstable or under-damped. Better performance and rejection of interference is shown when the frequency measurement uses data which does not include the phase calibrations for hardware (such as anti-alias filter response). On the other hand, the PMU outputs such as the measured amplitude and phases (both single-phase and positive-sequence) need to account for this calibration data.

In this algorithm, if the input frequency f is constant then the algorithm will behave in an identical manner to Symmetric algorithm. However, when the input frequency f is changing, the loopback frequency f_M* changes continuously as it tracks f, and therefore both the correlation frequency and the filter weights/notches also change on a continuous basis.

One effect of this is that while it still makes sense to position the timestamp at the centre of the filter time window, as per (3), the value of Φ_C at the timestamp cannot be exactly deduced simply by backtracking using the most recent values of f_M* and half the filter time window length, and (4**)** no longer holds. In fact, for the short 2-cycle P-class filter, the error which would be incurred in using (4**)** would be very small, but the error which would be incurred for a longer M-class filter would be significant. This is the similar effect of an input frequency chirp as was discussed above for the symmetric algorithm. However, for the asymmetric filter the correction of (4**)** is relatively easy to perform. Since the values of Φ_C are known at every computational frame, it is possible to hold these values in a memory buffer and retrieve the value from a time exactly $1/f_C$ seconds (half the filter time window) in the past, using linear interpolation between samples where required. This process automatically corrects for the frequency-chirp effects when the input frequency f (and also the loopback frequency f_M*) are rising or falling.

One effect of this is that there is a feed-forward effect within the frequency calculation. This is because the value of Φ_C is incremented every computational frame by f_M*T_s , where T_s is the frame time (reciprocal of the sample rate). Also, during steady changes in f where ROCOF is constant, both f and f_M* rise at the same rate. The properties of the chirping correlation and filter then lead to the surprising result that there is essentially no latency in the initial frequency measurement during times that ROCOF is constant. (But during changes in ROCOF, there is the expected latency). Consequently, to generate a measurement of frequency which is accurate at the timestamp, for constant-ROCOF situations, the frequency measurement needs to be delayed by half the filter time window length. More usefully, this "spare time" can be used to apply a further averaging over exactly 2 cycles. This reduces general noise, and the averaging over exactly 2 cycles places further filter notches at all multiples of $1/(2f_M$*) Hz. This "bonus" filtering is extremely beneficial for the PMU frequency and ROCOF outputs.

IV. BENCHMARKING

To verify the suitability of the algorithms for real-time implementation, they have been benchmarked on two different processors. The Infineon TC1796 microcontroller which has relatively slow RAM (Random Access Memory) access and small RAM size, and the Motorola MVME5500 which has a large RAM size and fast RAM access. All algorithms were configured for 2.5kHz sampling (a frame time of 400µs). The algorithms are coded in MATLAB® Simulink and converted to 'C' code using the RealTime Workshop and Embedded Coder toolboxes. The execution times were then measured as described in [11]. The results in Table I show that, while seemingly much more complex than the "Basic" PMU, the asymmetric PMU algorithm executes about 5 times faster. This is primarily because while the standard FIR filter in the "Basic" algorithm needs to evaluate the multiplications and additions across the entire time window at every computational frame, the cascaded averaging filters only need to evaluate the changes to the filter outputs due to the incoming and outgoing samples at the ends of the time window. Thus, as the sample rate is increased, the number of calculations in the averaging filters is not increased (but the memory requirement of the buffers is) [9, 10].

Based on the measurements in Table I, there would be no problem in executing either of the proposed algorithms at a 10kHz sample rate (100µs sample time). This removes the risk of aliasing of harmonics up to the 50th and well beyond.

TABLE 1 : BENCHMARKING OF ALGORITHMS

P-class benchmark results	**TC1796 results**		**MVME5500 results**
	Execution time	***RAM requirement (32-bit precision)***	***Execution time***
Basic	180µs	2.4 kB	81µs
Symmetric twin filters in "tick-tock"	54µs	20 kB	24µs
Asymmetric filter	33µs	13 kB	17µs

V. RESULTS

A test environment has been created in software which exposes the algorithms to a sequence of waveforms and interfering effects. The PMU algorithm outputs can be compared to the known synthesized signal properties (frequency, ROCOF, amplitude, phase) by careful post-analysis, taking into account the timestamps applied by the PMU algorithms. The test suite presented is not designed to provide formal demonstration of compatibility with C.37.118 (this activity is being carried out as part of the EURAMET EMRP programme [13]), but provides a practical demonstration of relative PMU performance.

The sample rate used for the PMU algorithms is 10 kHz. Although PC.37.118 suggests that no anti-alias filter should be used for P-class devices, a first-order 2.5kHz low-pass filter has been modelled (and is calibrated for by the PMU algorithms). A 12-bit ADC (analogue to digital converter) is modelled, with a Gaussian noise of 0.1-bits RMS. The waveforms and filter are simulated at a time step of 25µs, taking care to model the anti-alias filter so that its latency in the simulation domain matches the latency of an analogue filter very closely (since even 25µs represents 0.45° at 50Hz).

A. Scenario 1 : High levels of unbalance and harmonics

In the first scenario presented, the waveforms contain severe unbalance, harmonics to 27% THD, and inter-harmonics. These are more difficult challenges than the draft PC.37.118 standard describes. However, such waveforms are potential real scenarios within low voltage distribution grids, particularly for measurements of current, or during transient changes.

The waveform inputs for the first test case are:

- t=1s to t=2s: 50Hz, 1-per-unit (pu), clean sinusoids
- t=2s to t=3s: Ramp to 52Hz at a ROCOF of 1 Hz/s
- t=3s to t=4s: Add 5th Harmonic to all phases at 10%
- t=4s to t=5s: Harmonics 2 to 40 at random phases (but balanced on all three phases), total THD 28%
- t=5s to t=6s: Add unbalance at 2% (negative sequence 0.02pu, positive sequence drops to 0.98pu)
- t=6s: Instant phase jump of 20°
- t=7s to t=7.5s: Dip phase A to 10% by reducing positive sequence to 0.7pu, and adding -0.3pu negative and -0.3pu zero sequence.
- t=8s to t=9s: Add unbalanced inter-harmonics at 525 Hz (10%) to all three phases.

Figure 8 shows the TVE (Total Vector Error) for the three algorithms. This shows that all three algorithms have similar and compliant TVE performance if calibrated correctly, with errors greater than 0.1% only occurring transiently after step changes to interfering qualities, which is compliant with the standard under the "response" and "delay" time allowances.

Figure 9 shows the errors on the frequency measurements from the 3 algorithms. Clearly, the "Basic" algorithm gives some large frequency errors, because the fixed-weight filters do not filter out the interference caused by harmonics and unbalance, when the fundamental frequency is off-nominal at 52 Hz. The performance of the proposed algorithms is far superior. The Asymmetric filter algorithm shows ringing on the frequency measurement immediately subsequently to step changes in the interfering qualities (harmonics, unbalance, ROCOF), but provides the best results when these interfering qualities are steady.

The ROCOF PMU outputs follow similar trends since they are derived directly from the frequency output. Notably, the ROCOF output for the "Basic" algorithm in this scenario is unusable, containing noise to 100 Hz/s or more from t=3s onwards (Figure 10). The ROCOF for the proposed two algorithms is also shown in Figure 10, clearly showing that the asymmetric filter algorithm is the best. The asymmetric filter is better at reducing noise due to harmonic and random perturbations than the symmetric filter, due to the additional 2 cycles of averaging which can applied to the frequency and ROCOF measurements using this algorithm (see section IIIB).

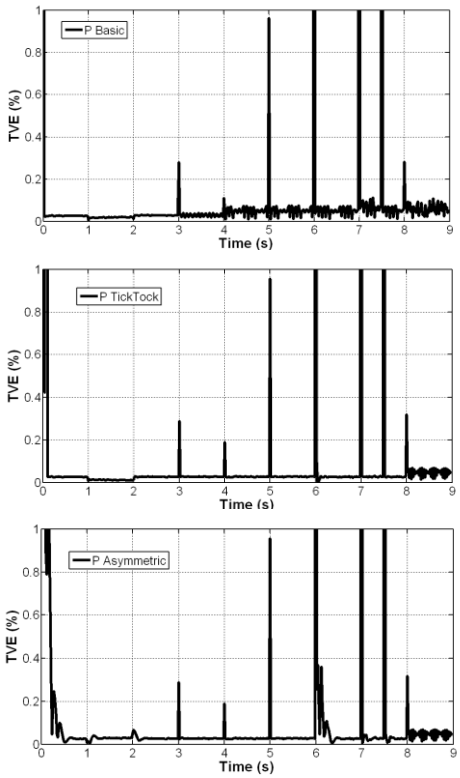

Figure 8. Scenario 1 : Total Vector Error (TVE) errors in % for PMU algorithms: Basic (top), Symmetric "tick-tock" (middle) and Asymmetric (bottom)

To accurately compare the phase angles reported from two PMUs, not only the vectors and timestamps need to be known accurately, but so do the frequency and ROCOF measurements. This is because the PMU measurements may have different timestamps when there are compared, due to communication delays. In these cases, the phase angles need to be corrected by formulas based upon the following equation:

$$AngleCorrection(Degrees) = 360\left(f \cdot \Delta t + \left(\frac{df}{dt}\right)\frac{(\Delta t)^2}{2}\right) \quad (6)$$

where Δt is the time elapsed since the measurement timestamp, f is the measured frequency, and df/dt is the measured ROCOF.

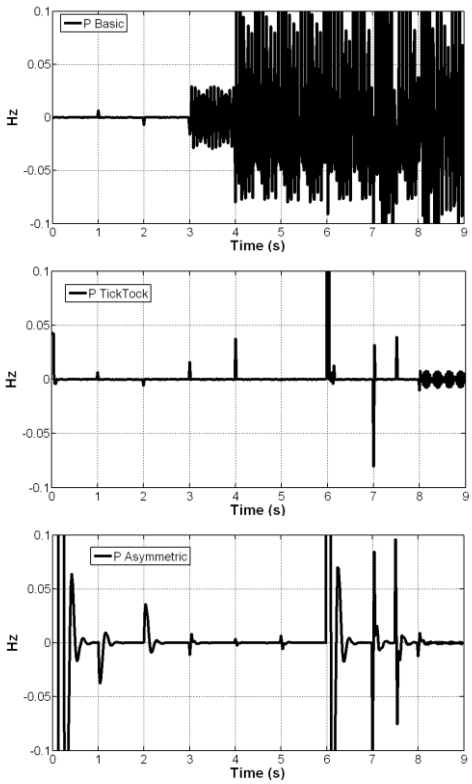

Figure 9. Scenario 1 : Frequency errors for PMU algorithms: Basic (top), Symmetric "tick-tock" (middle) and Asymmetric (bottom)

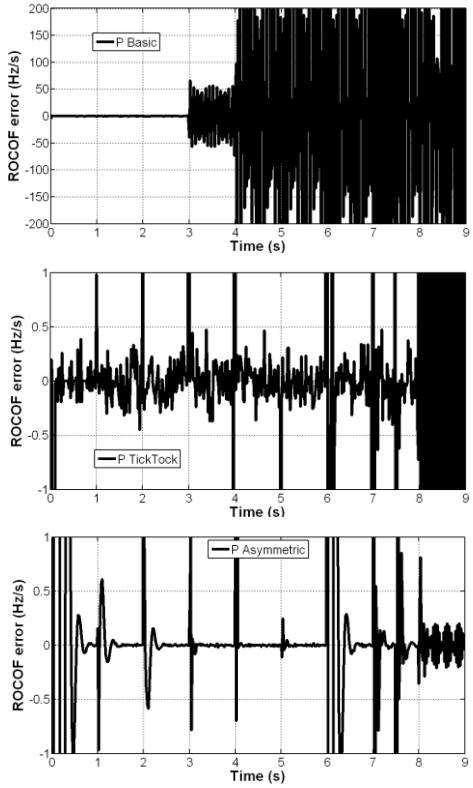

Figure 10. Scenario 1 : ROCOF errors for proposed PMU algorithms: Basic (top), Symmetric "tick-tock" (middle) and Asymmetric (bottom)

Assuming a 100ms communication delay between two PMUs needs to be compensated, and that an angle accuracy of 0.5° is required (a TVE of 0.9%), then even if the original TVE measurement is exactly accurate, the maximum tolerable frequency and ROCOF errors by (6**)** are in the region of 0.01 Hz and 0.3 Hz/s. This suggests that perhaps the value of ±0.01 Hz/s specified in [5] is inappropriately small and should be increased.

B. Scenario 2 : Balanced 1% 5th Harmonic

Since Figure 8 to Figure 10 show such high values of frequency and ROCOF error, particularly for the "Basic" algorithm, this second scenario carefully examines these parameters under less harsh conditions specified explicitly by [5]. The frequency error specification is ±0.01 Hz during frequency ramps of up to ±1 Hz/s, and ±0.005 Hz at steady state. The ROCOF error specification is only ±0.01 Hz/s. At steady state with frequency in the range f_0±2 Hz, 1% of any harmonic up the 50th may be present.

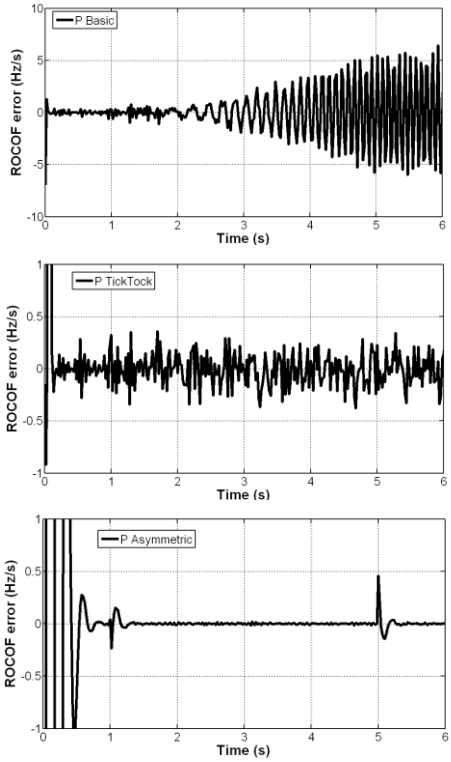

Figure 11. Scenario 2 : ROCOF errors for proposed PMU algorithms: Basic (top), Symmetric "tick-tock" (middle) and Asymmetric (bottom)

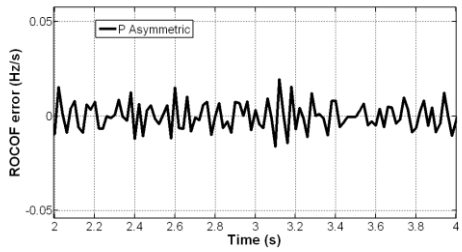

Figure 12. Scenario 2 : ROCOF errors for Asymmetric PMU algorithm.

Figure 11 shows the results with a steady application of a balanced 5th harmonic at a relative magnitude of 1%, but no other departures from balanced sinusoidal waveforms. Frequency starts at 50Hz for the 1st second, but then ramps from 50-52 Hz over the time between t=1s and t=5s (a 0.5 Hz/s ramp), and is then held steady at 52 Hz for t=5s to t=6s.

Clearly the ROCOF errors given by the "Basic" algorithm are well outside the specified limits when frequency is off nominal, and also far above a sensible value of 0.3 Hz/s which would enable comparison of the results from 2 PMUs, accounting for communication delays. The ROCOF errors from the Symmetric "tick-tock" algorithm are much lower, but still well above ±0.01 Hz/s. The errors from the Asymmetric algorithm are actually marginally compliant, as shown by the magnified trace in Figure 12. All three algorithms are compliant with the TVE and frequency specifications (1% and ±0.01 Hz) in this scenario.

VI. CONCLUSIONS

The "Basic" algorithm described in PC37.118 is easily capable of meeting the magnitude, phase and TVE (total vector error) measurement specifications during frequency ramps and under the influence of unbalance, harmonics and inter-harmonics at quite high levels. However, the measurements of frequency and ROCOF from this "Basic" algorithm have been shown to be very sensitive to the presence of harmonics and unbalance when frequency is off-nominal. This is due to the inability of the "Basic" FIR filters to adapt to the off-nominal frequencies and filter out unwanted mixing frequencies from the Fourier correlation.

In particular, the "Basic" algorithm cannot meet the stated ±0.01 Hz/s ROCOF accuracy under the influence of just a single 1% 5th harmonic. In fact, its ROCOF errors are up to 6 Hz/s at 52 Hz. The validity of the ±0.01 Hz/s specification in PC37.118 is questioned in this paper, and instead a value of about 0.3 Hz/s is perhaps more sensible for P-class PMUs.

This paper describes two algorithms which both perform significantly better than the "Basic" algorithm at measuring frequency and ROCOF during complex scenarios involving harmonic interference and unbalance when frequency is off-nominal. The two algorithms have different properties. The symmetric filter algorithm is more complex, having two parallel filter paths, and requires almost double the memory of the asymmetric filter algorithm. The asymmetric filter algorithm shows a tendency to "ring" following transients, but once settled, the frequency and ROCOF errors are significantly lower than those of the symmetric algorithm. This might be extremely useful since any device receiving and comparing data from multiple PMUs with different timestamps will need to use the frequency and ROCOF measurements to de-skew the PMU measurements so that they can be compared, and therefore frequency and ROCOF errors will increase the perceived phase error at the comparing device.

It might be possible to create a hybrid of the two proposed algorithms, to provide a measurement with the minimal ringing of the symmetric filter algorithm, but the best frequency and ROCOF performance of the asymmetric filter algorithm.

REFERENCES

[1] D. M. Laverty, D. J. Morrow, R. Best, and P. A. Crossley, "Performance of Phasor Measurement Units for Wide Area Real-Time Control," *2009 IEEE Power & Energy Society General Meeting, Vols 1-8,* pp. 4282-4286, 2009.

[2] M. G. Adamiak, A. P. Apostolov, M. M. Begovic, C. F. Heriville, K. E. Martin, G. L. Michel, A. G. Phadke, and J. S. Thorp, "Wide area protection - Technology and infrastructures," *IEEE Transactions on Power Delivery,* vol. 21, pp. 601-609, Apr 2006.

[3] IEEE, "IEEE Standard for Synchrophasors for Power Systems," C37.118-2005, 2005.

[4] K. E. Martin, D. Hamai, M. G. Adamiak, S. Anderson, M. Begovic, G. Benmouyal, G. Brunello, J. Burger, J. Y. Cai, B. Dickerson, V. Gharpure, B. Kennedy, D. Karlsson, A. G. Phadke, J. Salj, V. Skendzic, J. Sperr, Y. Song, C. Huntley, B. Kasztenny, and E. Price, "Exploring the IEEE standard C37.118-2005 synchrophasors for power systems," *IEEE Transactions on Power Delivery,* vol. 23, pp. 1805-1811, Oct 2008.

[5] IEEE, "IEEE Draft Standard for Synchrophasors for Power Systems," PC37.118.1/D3.0, 2011.

[6] G. Y. Yang, K. E. Martin, and J. Østergaard, "Investigation of PMU Performance Under TVE criterion," in *5th International Conference on Critical Infrastructure (CRIS)*, 2010.

[7] A. G. Phadke and B. Kasztenny, "Synchronized Phasor and Frequency Measurement Under Transient Conditions," *IEEE Transactions on Power Delivery,* vol. 24, pp. 89-95, Jan 2009.

[8] R. Petrella, A. Revelant, and P. Stocco, "Robust Grid Synchronisation in Three-Phase Distributed Power Generation Systems by Synchronous Reference Frame Pre-Filtering," in *Universities' Power Engineering Conference (UPEC)*, Glasgow, UK, 2009.

[9] A. J. Roscoe, R. Carter, A. Cruden, and G. M. Burt, "Fast-Responding Measurements of Power System Harmonics using Discrete and Fast Fourier Transforms with Low Spectral Leakage," in *1st IET Renewable Power Generation Conference*, Edinburgh, Scotland, 2011.

[10] A. J. Roscoe and G. M. Burt, "Comparisons of the Execution Times and Memory Requirements for High-speed Discrete Fourier Transforms and Fast Fourier Transforms, for the Measurement of AC Power harmonics," in *2nd IMEKO TC 11 International Symposium METROLOGICAL INFRASTRUCTURE*, Cavtat, Dubrovnik Riviera, Croatia, 2011.

[11] A. J. Roscoe, S. M. Blair, and G. M. Burt, "Benchmarking and optimisation of Simulink code using Real-Time Workshop and Embedded Coder for inverter and microgrid control applications," in *UPEC: 2009 44th International Universities Power Engineering Conference*, 2009, pp. 532-536.

[12] A. J. Roscoe, G. M. Burt, and J. R. McDonald, "Frequency and fundamental signal measurement algorithms for distributed control and protection applications," *IET Generation, Transmission & Distribution* vol. 3, pp. 485-495, May 2009.

[13] EURAMET EMRP, "EMRP Call 2009 ENG04 Metrology for smart grids," 2010 Available: www.euramet.org/index.php?id=emrp_call_2009#c8319, accessed October 2011.

.