# Implementation and Performance of Synchrophasor Function within Microprocessor Based Relays

**Bogdan Kasztenny**
**GE Multilin**

**Mark Adamiak**
**GE Multilin**

## 1. Introduction

Synchronized phasor measurements have come a long way since their conception [1]-[3]. Many potential applications have been identified [2], including improved state estimation, frequency estimation, instability prediction, adaptive relaying, and wide area control, for example. The recently published IEEE Standard C37.118 [3] will assure that compliant phasor measurement units will all report phasors using the same convention for measuring phase angle, particularly when the underlying power system frequency is off-nominal.

Even though today's installations of Phasor Measurement Units (PMUs) are limited in cover-age and enterprise communication performance, it is clear the technology will advance quickly yielding significant benefits.

Widespread deployment of the PMUs providing for both appropriate penetration and redundancy of synchronized measurements is a key factor. Such widespread deployment can be achieved when integrating the PMU function within modern microprocessor-based relays - similar to the relay integration trend seen with metering, fault recording, and sequence of event re-cording capabilities.

Implementation of PMU functions, however, imposes new requirements on protection plat-forms. Most importantly PMUs require correlation of the waveform samples with the absolute time driven by the Global Positioning System (GPS), and reporting the measured phasors with reference to such absolute time. Traditional relay implementations sample their input voltages and currents asynchronously from any external process such as the GPS time, and derive measurements as quickly as possible for speed of response.

Additional processing requirements are presented for the relay to measure, communicate and record the PMU data in addition to providing for their core protection functionality.

All this raises potential concerns with respect to integrating PMU functions in protective re-lays. Whether on new platforms or as a part of an upgrade of an existing product, changes to the existing or commonly deployed data acquisition system of a relay in order to accommodate synchrophasors can have serious consequences for all the other functions of the relay- protection in particular.

This paper reviews the basic aspects of synchrophasor implementations integrated with protective relay functionality. It presents the key technical challenges, and discusses solutions that eliminate the risk of impacting the core protection functionality of the relay. The paper offers simple tests that can be applied to gauge the impact of an integrated PMU on the overall performance of a given relay.

The overall goal of this paper is to educate the user and allow for more rational decision making with respect to deploying integrated PMUs versus standalone PMUs.

## 2. Issues when Implementing Synchrophasors on Protection Platforms

It is self evident that wide penetration of PMUs facilitating both faster accumulation of experience in preparation for advanced applications, and redundancy of measurements required for the future critical applications of synchrophasors, can be naturally achieved by integrating PMU functions with protection and control platforms. Successful integration of sequence-of-events (SOE) and digital-fault-recorder (DFR) capabilities with protective relays is a historical lesson to follow when considering cost-efficient and widespread deployment of PMUs.

Modern protection platforms are capable of supporting synchrophasor measurements, local re-cording and reporting. This relates to internal architectures, time synchronization, metering accuracy, communication capabilities, and processing power required to comply with the C37.118 requirements.

However, microprocessor-based protection relays have been designed historically without regard to the notion of absolute time. Time stamping for SOE and DFR recording is probably the only instance of reference to an absolute time in protective relaying. Sampling and synchronization, even in critical and high performance systems such as the line current differential protection, is typically achieved without reference to the absolute time. This is a prudent protection approach as it limits exposure of mission-critical protection functions to availability and misbehavior other devices such as the GPS system and associated receivers/clocks.

Predominantly protection relays sample asynchronously with respect to the absolute time, but in synch with power system frequency. The latter is to keep the digitally implemented measurements accurate should the power frequency depart from its nominal value.

The following sections provide some insight on implementation of synchrophasors on a typical microprocessor-based relay [4]. It presents some solutions, and highlights certain aspects that need to be understood and evaluated by a protection engineer to make sure the extra functionality put on a relay does not jeopardize the core protection task of the device.

## 3. Design Principles when Implementing Synchrophasors on Existing Platforms

It is prudent to follow these design principles when implementing synchrophasor measurements on existing or new protection platforms:

1. The underlying sampling process of the relay shall not be altered. Sampling and data collection potentially affects all other functions of the relay. To minimize the risk, this area shall not be modified. Sampling in synchronism with the absolute time is not only unnecessary; it actually yields a substandard solution from the point of view of metering accuracy as shown later in this paper.

2. The synchrophasor calculations shall be added in parallel to the existing protection, control and metering functions to minimize the risk of affecting these critical functions.

3. Hardware modifications shall be minimized for the reason of stability of the design.

4. Calculations shall be organized in a way that the extra processing power is optimally distributed and can be accommodated by existing platforms with appropriate security margin, even under fault conditions and other periods of increased activity of an IED.

The key design areas for implementation are: timing accuracy; sampling and correlating input signals with the absolute time, algorithms for accurate measurement of the phasors, data storage, recording and streaming.

## 4. Timing Accuracy

Accuracy of synchrophasors as measured by the C37.118 is defined as a Total Vector Error (TVE) being the percentage magnitude of a vectorial difference between the measured and actual phasors treated as vectors.

As such the TVE has three major components: magnitude error, angle error as related to the input signals, and angle error as related to the measurement of the absolute time.

It is enlightening to think of time as a quantity that needs to be "measured" by a given device based on a standard physical input, such as the 1 pulse per second (1pps) marker embedded in the standard IRIG-B input. Assuming 1% TVE target as per the C37.118, and budgeting accordingly for the three sources of error, leaves up to 5-8 microseconds for the total timing error.

Not only does a given device needs to synch with the 1pps signal but between the pulses, the device must internally maintain a very precise notion of time so that each of the synchrophasor reference points (referred in this paper as "synchrophasor interrupts") occurring within the period of the full second is maintained with an error not larger than few microseconds.

Figure 1 illustrates this process. In one particular implementation a precise phase lock loop is run with the objective to null out the positional error between the 1pps signal and the last synchrophasor interrupt that ought to occur exactly at the top of the second. This phase lock loop compensates for the natural drift of the internal IED oscillator, and the finite resolution of the latter.

For example, a given oscillator could have an error of say 25 parts per million as specified by the component manufacturer. This means it could drift up to 25 microseconds over a period of 1,000,000 microseconds (1 second). This value would prevent successful implementation of synchrophasors. Moreover, the error can change with temperature and between different articles (samples) of the oscillator. The drift, however, is easily measurable with the aid of the 1pps signal. When measured, validated, and averaged, the drift of the oscillator is an input to the phase lock loop making the internal time keeping mechanism extremely accurate.

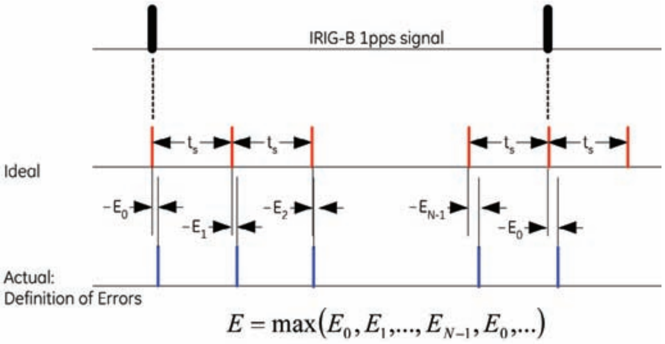

**Figure 1.**
*Defining synchrophasor interrupts and timing errors.*

The compensation process works as follows. The algorithm, using a precise hardware-implemented interrupt service, captures the local relay time (oscillator value) at the exact moment of the 1pps reference pulse. If the internal relay oscillator is perfect, the captured value should be exactly 1,000,000 microseconds from the last 1pps pulse. A reading of 1,000,015 microseconds, for example, means the oscillator is 15 microseconds / second too fast; while the reading of 999,994 microseconds means the oscillator is 6 microseconds / second too slow.

The value of the second-over-second drift is checked for validity, and averaged over longer periods of time. The secure and smoothed out value is now used to control the oscillator or as a correction in the algorithm generating the synchrophasor interrupts. Our solution uses the measured drift of the internal oscillator to discipline the synchrophasor interrupt generator rather than control the oscillator [4]. This avoids changes to the relay and thus following the design goals outlined in section 3 above.

Another issue is the required resolution of the internal oscillator. Assuming 60 synchrophasors are produced per second, the synchrophasor interrupts are to be generated every 1,666.66(6) microseconds. When this number is rounded to a practical oscillating frequency, an error would accumulate making the last synchrophasor interrupts in a given second inaccurate. For example assume an implementation using 0.25-microsecond resolution that is generating synchrophasor interrupts every 1,666.50 microseconds. The 1,666.50 microsecond interval is off only by 0.166(6) microseconds from the ideal value. However, after 1 full second when this error adds up 59 times, the last interrupt within the second will come after 60*1,666.50 = 999,990 microseconds that is a 10 microsecond error from the required time.

In addition, assume the oscillator is too slow by 12 microseconds in each second (example). To compensate for the drift each synchrophasor interrupt will have to be adjusted by 12/60 = 0.2 microsecond, while a practical resolution of the oscillator can be in the range of a quarter of a microsecond. The 0.25 – 0.20 = 0.05 microsecond error repeated 60 times within each second would yield 3 microseconds of error eating away from the tight timing error budget required by synchrophasors.

To minimize this error, a dithering algorithm is applied yielding a high accuracy of timing for the synchrophasor interrupts. An internal variable is used to count the time with a nanosecond accuracy, while the interrupts are generated with a 0.25 microsecond resolution. The device keeps track of the error accumulated due to the finite resolution of the oscillator. Once the error reaches half the resolution period, the synchrophasor interrupt is moved by one resolution period. In this way the error is kept below half the period of the oscillator, and never accumulates.

The discussion on timing presented in this section is an excellent illustration of issues and challenges faced when implementing synchrophasors on existing relay platforms or traditionally designed new relay platforms. The solutions outlined in this section are elegant and avoid any changes to the existing relay hardware, thus minimizing the risk and avoiding expensive internal oscillator upgrades. The “time keeping” is implemented in software based a on carefully crafted algorithm.

## 5. Sampling for protection and synchrophasors

Protective relays typically do not sample synchronously with respect to the absolute time. In-stead, they sample based on a free-running sample and hold timer and often apply frequency tracking or compensation so that the measurement calculations retain accuracy even if the system frequency departs from the nominal value. It is a common misconception that measuring synchrophasors requires sampling synchronously to absolute time.

Some applications force the data acquisition system (A/D converter) to take samples at precise pre-defined points in time with respect to absolute time. This, however, results in unnecessarily complicated designs, and is not required. In order to measure synchrophasors one needs to know the absolute time of each sample taken by the A/D, but these samples can be taken at any point in time. They do not have to be “hard-synched” to the GPS clock.

Relays and other devices measuring sine waves apply frequency tracking. These devices calculate features of sine-waves (magnitude, for example) using their measuring algorithms such as the Fourier Transform. These algorithms assume typically a constant pre-defined number of samples taken in each period of the waveform. If the system frequency changes, the period changes, and the number of samples in a period would change as well if using a constant sampling rate. This would yield certain finite measurement error. In order to eliminate this error either the sampling rate is made variable to follow the system frequency, or a numerical compensation is programmed in the device.

The first approach is typically more popular and referred to as “frequency tracking”. Effectively, frequency tracking varies the length of the data window used for digital measurements to follow the length of the signal period as it varies under off-nominal frequencies and power swings.

Another misconception is that staying in synchronism with the system frequency (for accuracy) and staying in synchronism with the absolute time (for phase reference) are contradicting targets, and require convoluted solutions such as measuring the magnitude and angle using different algorithms.

The former is about adjusting the length of the data window so that it covers pre-selected multiplies of power cycles; the latter is about positioning of this data window so that the measurement complies with the C37.118 angle convention.

Both can be controlled independently with no major obstacles. One may think about these two processes as having two controllers: one positions the center of the data window to align it precisely with the synchrophasor interrupts; the other controls the sampling rate to keep the length of the data window in relation to the slightly changing system frequency.

Although the samples must be correlate-able to absolute time, they can be taken at any time instant. Figure 2 presents a solution in which the samples are collected asynchronously with respect to absolute time. The platform applies frequency tracking to keep the number of samples constant in the actual period of the waveform as the period changes [4]. When the synchrophasor interrupt is asserted, the device locks the sample index and collects half its data window from the samples that follow the interrupt and half – from the samples preceding the interrupt. In this way, without altering the sampling process the device gets a data window that is placed very closely with respect to the required reporting point in time.

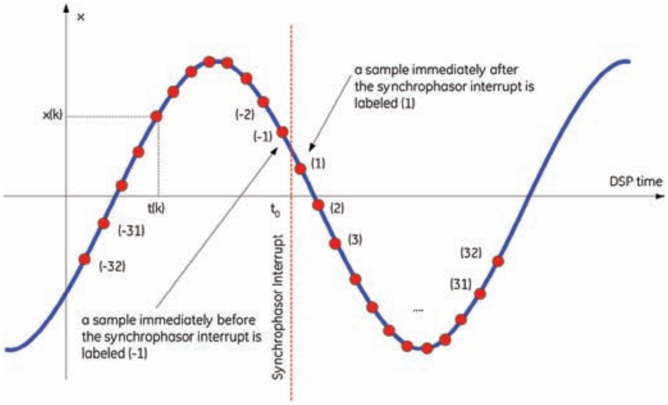
x
x(k)
t(k)
(-2)
(-1)
(1)
(2)
(3)
(-31)
(-32)
(31)
(32)
....
$t_0$
DSP time
Synchrophasor Interrupt
a sample immediately after the synchrophasor interrupt is labeled (1)
a sample immediately before the synchrophasor interrupt is labeled (-1)

**Figure 2.**
*Data window based on asynchronously taken samples.*

Note that in this approach:

- The length of the data window is already correct and adequate as the sampling period is controlled by the frequency tracking mechanism;
- The position of the window is within half of the sampling period from the required position as per the synchrophasor convention.
- It is trivial for the device to calculate the offset between the center of such "best-placed" window and the required position of the window. Calculate of this difference does not require referring to absolute time. This time difference is used to compensate the synchrophasor measurements as explained below.

The device calculates the center of the window by averaging the time stamps of the samples within the window. This averaging is done using any time reference, not necessarily the absolute time reference. In our implementation a free running microsecond counter is used to calculate the position of the center of the data window. The same free running counter is used to capture the time of the synchrophasor interrupt asserted based on the true absolute time. Even though the free running microsecond counter is not a true time, the time difference between the synchrophasor interrupt (point when the center of the window should be), and the calculated center of the window (point when the data window actually is) is precise and can be used for compensation.

Following the window selection procedure illustrated in Figure 2 the DSP places the window to within few degrees to the synchrophasor interrupt. The inherent displacement is precisely measured and is used for very precise compensation of the calculated phasor (an angular rotation of 2-3 degrees as described later in this paper).

This approach is ideal for typical relay architectures: samples are taken by data acquisition systems typically incorporating an A/D converter and a Digital Signal Processor (DSP). These data acquisition subsystems typically do not have a notion of absolute time. In our approach a very simple solution is adopted. In this architecture (Figure 3) the Central Processing Unit (CPU) of the IED synchronizes to the 1pps signal and executes the phase lock loop that generates precise synchrophasor interrupts. These interrupts are captured by the DSP using a "local DSP time" in the form of a free running counter. The interrupt triggers calculations for the synchrophasor instant and allows the DSP to obtain the notion of time, and produce the phasor precisely aligned with the time mark as driven by the interrupt.

## 6. Post-processing and Extra Filtering

As depicted in Figure 4, our device uses "best-placed" windows for synchrophasor measurement without altering the sampling process (windows X). It measures the small shift between the required and actual positions of such windows and compensates for the difference by a simple phasor rotation. This yields synchronized full-cycle Fourier windows (windows Y).

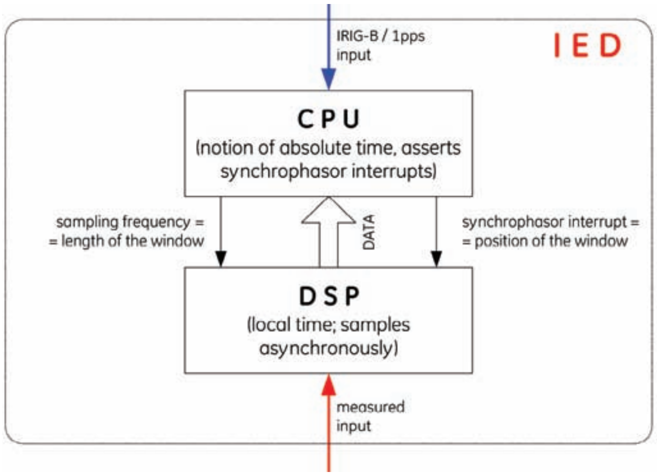
IRIG-B / 1pps input
IED
CPU
(notion of absolute time, asserts synchrophasor interrupts)
sampling frequency = = length of the window
DATA
synchrophasor interrupt = = position of the window
DSP
(local time; samples asynchronously)
measured input

**Figure 3.**
*CPU & DSP architecture for synchrophasor implementation.*

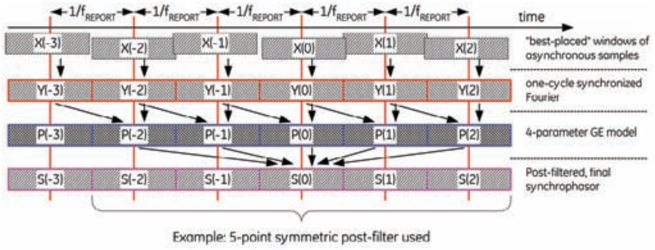

**Figure 4.**
*Processing of best-placed raw data windows into synchrophasor values.*

The X and Y windows are produced at nominal system frequency regardless of the recording or reporting rates set for the PMU function. A pair of Y windows (the present and past windows) is used to implement the four-parameter signal estimator as described later in this paper. As a result a new, more accurate estimate of the phasor is calculated at the rate of nominal system frequency (windows P in Figure 4). The P-values are calculated assuming the phasor may change in time, and as such are extensions of the C37.118 synchrophasor standard, aimed at future dynamic applications of synchrophasors.

In order to control the balance between speed and accuracy of the measurement, the device further implements user selectable post-filtering, that is, a number of P-measurements can be combined into the filtered synchrophasor output, S, effectively extending the estimation window. The post-filtering is not a straight average, but takes into account the value and rotation speed of each of the used P-values as described later in this paper.

## 7. Compensating for Analog Errors

Synchrophasor implementation calls for accuracy above a typical protection accuracy or metering accuracy as typically provided on protective relays. When implemented on a protection platform, synchrophasors may need correcting for errors of the IED's input transformers.

Figure 5 shows a correcting function for the current inputs: the correction is small – in the order of 0.2 – 1.8 degrees – and depends on both the magnitude and frequency of the signal. In particular at very low signal levels and lower frequencies the excitation current of the input trans-formers starts causing some angular errors, and the device applies higher correction for the measured angle for the current inputs.

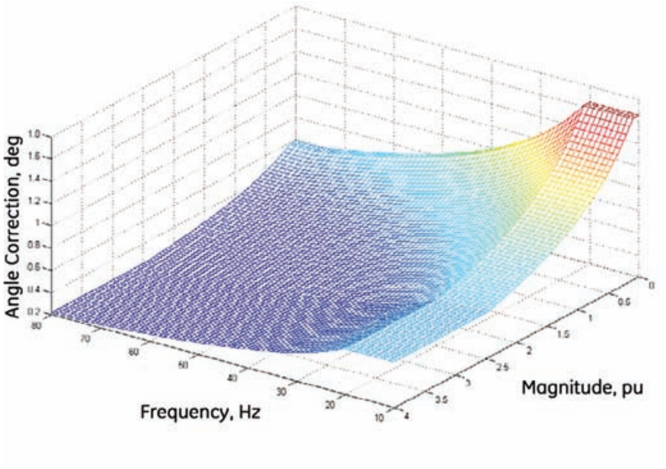

**Figure 5.**
*Correction of current input transformers.*

Figure 6 shows the correction applied to the voltage inputs. The required angle shift to keep the measurements accurate is smaller (up to 0.2 degrees), and again depends on the magnitude and frequency of a given voltage input.

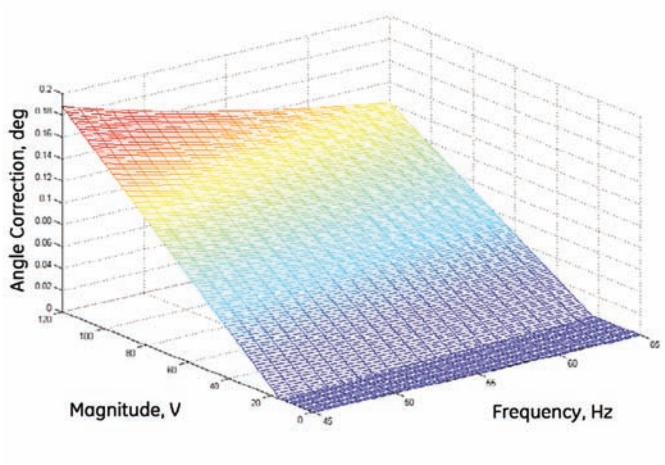

**Figure 6.**
*Correction of voltage input transformers.*

Analog filters, necessary in any digital measuring system to deal with aliasing of samples, introduce a phase shift, which also needs to be compensated. When the analog filter is set relatively high, the phase shift for the frequency band around the nominal is very linear, and can be easily compensated. Figure 7 shows the measured (red dots) and applied (blue line) correcting angles accounting for the impact of analog filters in the solution [4].

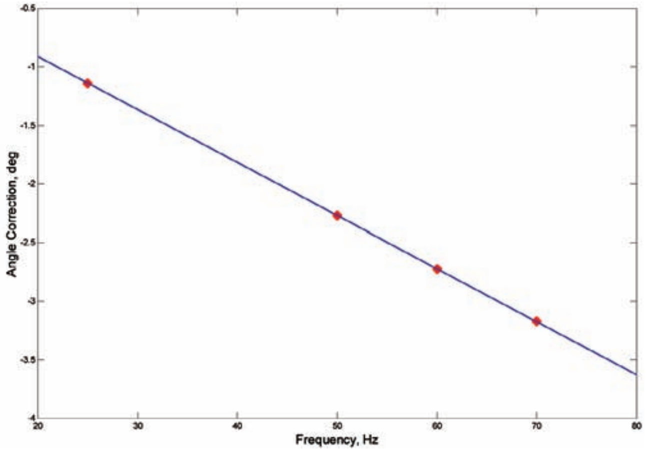

**Figure 7.**
*Correction for the Analog Filter.*

The few implementation details outlined above are meant to direct attention to the way synchrophasors are implemented on protective relays, and the potential impact on the existing mission-critical protection functionality. In the outlined implementation minor hardware changes were required to provide synchrophasor interrupts from the central processing unit, having the notion of the absolute time (from the IRIG-B input), to the digital signal processor, which is responsible for the majority of the calculations but has no direct relationship with absolute time. All the other aspects of the synchrophasor implementation have been accommodated in software, in subroutines completely detached from the key protection functions. This minimizes the risk and allows claiming a very secure implementation [4].

## 8. Implementation of the Data Communication Protocol

An important part of the C37.118 synchrophasor standard is the interoperable data communication protocol. The C37.118 protocol is a low-overhead "lean" protocol well suited for real-time data communication. The communication is organized around 4 types of frames:

- Configuration frames describing either present or maximum device configuration are sent to the higher order system (Phasor Data Concentrator, PDC) on demand or automatically upon configuration change of the PMU. These frames are therefore sent only exceptionally and are intended for the PDCs.

- Header frame is similar to the configuration frames but is not standardized and contain human-readable information about the PMU.

- Command frame is sent by a PDC and received by the PMU (relay when integrated). Commands are sent to stop and resume data transmission, request configuration data, or execute actual commands by sending data that could be used to close/open an out-put and execute other user-programmable actions.

- Data frames are sent continuously by the PMU (relay) at regular and C37.118 standardized time intervals.

Streaming data frames is of primary concern when considering integrating PMUs with micro-processor-based relays. The standard specifies 30 frames per second as the fastest reporting rate, but some implementations support up to 60 frames/second. Data content may vary from a single phasor (typically the positive-sequence voltage) to several sets of three-phase voltages and cur-rents (frequency and rate-of-change of frequency are always sent).

Assume sending 6 phasors (3 currents and 3 voltages) each represented by 2 numbers (real and imaginary or magnitude and angle); with each number encoded on 2 bytes and reported at 30 frames a second. Ignoring the overhead one gets the bit rate of:

6 (phasors) × 2 (real, imaginary) × 2 (bytes) × 8 (bits) × 30 (frames / second) » 5.76kbps

Even when accounting for the protocol overhead and doubling the reporting rate, as increasing the packet size by including frequency, rate of change of frequency, etc. one stays within the DS0 level of 64kbps.

Modern protection relays are built to comfortably serve 64kbps real-time traffic. Such channels are used for teleprotection or in line current differential applications.

In additions, multi-function relays have been used for years to support SCADA and automation functions by providing for server functionalities of typical SCADA protocols (DNP, Mod-bus, UCA and IEC61850). Compared with these protocols the C37.118 synchrophasor protocol is neither complex nor demanding and can be safely implemented on a modern microprocessor-based relay.

## 9. Implementation of the Recording Functionality

Typically PMUs provide for data recording functions. These are useful in applications when no real-time communication is provided between the PMUs and the PDC, or in cases when the communication fails or is temporarily unavailable. Because system events are of interest, the time horizon for practical recorders is in the range of minutes or tens of minutes. This calls for mega-bytes of storage space.

Assume again the 5.67kbps data rate from the previous example, and consider a system event recorded for 10 minutes. The required storage space is in the range of:

10 (minutes) × 60 (seconds / min) × 5.67kbps » 3402kb, or 3.402/8 MB » 0.42MB.

Modern relay may provide for tens of MB of data storage, allowing records as long as few tens of minutes even at very high recording rates.

Proper engineering of the recording function needs to allow for the following:

- Safe recording for tens of minutes during which faults and other events can occur.

- Safe power down when recording – the control power can be removed when the PMU function is recording and producing massive records. No data corruption or other unexpected deficiencies should take place under such circumstances.

- Safe retrieval of stored data. When using a slow communication media to access the relay, it may take minutes to download the stored records. During this time faults, system events, or new records may occur. The relay needs to respond accordingly always giving priority to the protection functions.

Recording capabilities are standard on modern relays. The above problems exist today, and have been solved. The only difference between an existing fault recorder and an added PMU re-corder is the amount of data and duration of recording or extracting the record from the device.

## 10. Requirements for the Extra Processing Power

Measuring (calculating) synchrophasors including: precise timing, data collection and data processing, and various required corrections as described earlier; communicating the measured data as well as serving requests from the PDC; and performing local triggering and recording re-quire extra processing power.

Modern relays use multiple processors for data processing, logic engines, and communications. As a result it is achievable and safe to integrate the PMU function, assuming a prudent approach is taken with respect to the architecture.

In our solution each set of 8 analog signals (ac voltage and currents) is given a separate DSP to process the associated data. This results in a scalable architecture when adding more inputs to a given relay does not put more requirements on the DSP. This is no different with the synchrophasor calculations. When interrupted by the synchrophasor time tag, a DSP gathers a data window, calculates the full-cycle Fourier phasor, calculates the center of the window and the offset of the center with respect to the synchrophasor interrupt, compensates (rotates) the phase to account for the small offset, and compensates for the errors of VTs, CTs and the impact of analog filtering. These operations are very lean and account for only a small portion of the full set of typical DSP calculations required.

The rest of the process of calculating synchrophasors runs only at 60 times a second, and is relatively simple (Figure 4).

The communication protocol runs at up to 60 times a second, and therefore is relatively lean as well. The same applies to the integrated PMU recorder.

In our approach, the processing power required to provide for the PMU function even when reporting at the rate of 60 phasors a second, is at the level similar to calculations required to run one zone of distance protection. We consider it moderate and acceptable. No protection functions are suspended or delayed as a result of synchrophasor activities/calculations. No synchrophasor functions are suspended or delayed as a result of protection events or activities.

## 11. TVE Accuracy Achievable when Integrating PMUs on Protection Platforms

The following summarizes the steady state performance as tested on the IED hardware [4]:

- TVE for voltages, frequency range 45-70Hz< 0.30%
- TVE for currents, frequency range 45-70Hz< 0.40%
- TVE at 10% of THD, nominal frequency < 0.45%

Figure 8 presents results of the interfering frequency test when reporting at 60 times per second, and using a user-selectable 7-point post-filtering algorithm.

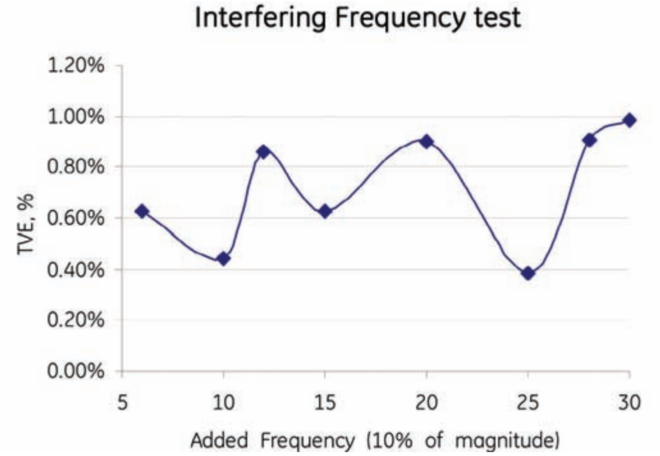

**Figure 8.**
*TVE under interfering frequency tests (reporting at 60/second, 7-point post-filter applied).*

The described implementation details, and the test results prove that when carefully engineered, modern P&C platforms allow for both secure and accurate implementation of synchrophasor measurement, recording and reporting. When integrated with protection platforms the PMU functionality is provided universally with wide coverage of the metering points, at a fraction of the cost of stand-alone PMU solutions.

## 12. Examples of Synchrophasors Measurements under Fault Conditions

This section presents few examples of synchrophasor measurements under simulated fault conditions.

Figure 9 shows a case of a reverse ABG fault as recorded by a line current differential relay. During the fault the system frequency was 59Hz, and the relay frequency tracking mechanism was intentionally disabled in order to test the response of both protection and PMU functions under frequency errors.

The top three traces show current waveform recorded by the relay. The next three traces are voltages, with the A and B voltages dropping to zero during the fault.

The last trace shown in Figure 9 is the operand of the 87L function. As expected, the integrity of this key function is not jeopardized by either the external fault, off nominal frequency, or PMU function operational on the same IED platform. Similarly other protection functions respond correctly. For example, the neutral directional reverse-looking overcurrent element picks up during the fault and stays operated for the entire duration of the fault.

The "PMU1 Va Mag" trace shows the magnitude of the phase A voltage as measured by the synchrophasor algorithm. The value is steady and accurate regardless of the off nominal frequency (signal at 59Hz, relay sampling at 60Hz). The "PMU 1 Va Angle" trace is the angle measurement. This value is recorded at 60 times / second and makes one full revolution every second. This is

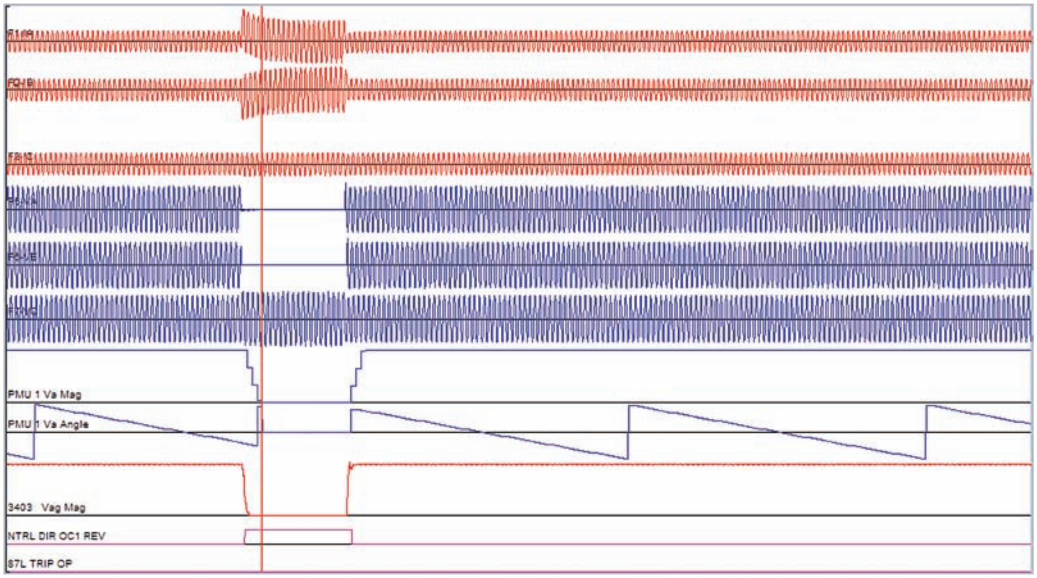

**Figure 9.**
*Sample record of a line-current differential relay containing both oscillography data (samples) and PMU data (synchrophasors).*

expected as the signal is at 59Hz, thus recorded 60 times a second it changes at (60-59)*360deg/sec.

For comparison the "3403 Vag Mag" trace is the voltage magnitude as measured for protection purposes. The synchrophasor version (PMU1 Va Mag) and the relaying version (3403 Vag Mag) are better shown in Figure 10. The synchrophasor measurement is implemented using an algorithm optimized for accuracy. As such this trace does not show the ripple distinctive for the off nominal frequency situation, and is accurate to within 1% of TVE. The protection measurement is affected by the off nominal frequency (visible ripple and the average value slightly off). This is because the relay was configured with frequency tracking disabled for the purpose of the test. Even with tracking disabled this particular relay shows only 2-3% of error in voltage for every Hz of frequency difference.

Figure 10 also illustrates that the synchrophasor values are recorded every 1/60th of a cycle (user setting), while the protection values are refreshed 8 times a cycle or every 1/480th of a second. Also, having less filtering and being optimized for speed rather than accuracy, the protection version of the voltage measurement responds much quicker to the voltage changes, exhibiting a short lasting overshoot when the voltage recovers after clearing this external fault. At the same time the synchrophasor measurement is very well controlled showing no overshoot or other problems.

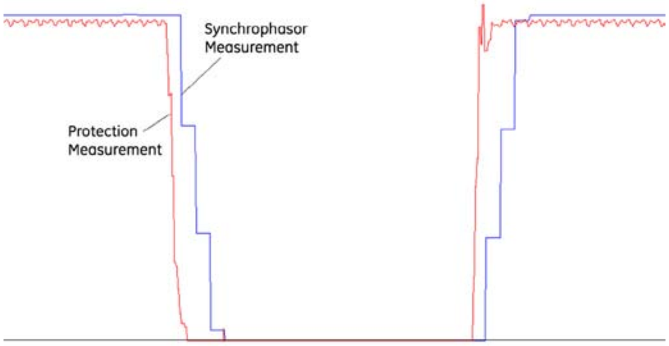

**Figure 10.**
*Synchrophasor and protection measurements on the same voltage signal in the record of Figure 9.*

Figure 11 shows and internal fault occurring under off nominal frequency (59Hz while the re-lay intentionally tracked to 60Hz). The fault is cleared by the 87L function as expected. Other protection, such as zone 2 shown in the Figure, operate as expected and stay picked up for the en-tire duration of the fault.

This test was done as a closed loop test resulting in opening the breaker. Once the breaker opened, the line-side VTs measure the voltage oscillating between the line capacitance and shunt reactors. The phase C voltage decays exponentially and the frequency measured by the relay changes from 59Hz in the pre- and fault periods, to about 50.3Hz being the resonating frequency between the line and its shunt reactors (Figure 12).

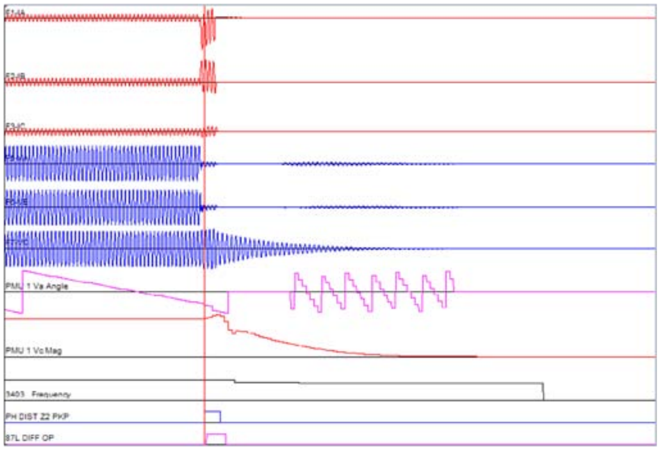

**Figure 11.**
*Sample record of a line-current differential relay containing both oscillography data (samples) and PMU data (synchrophasors).*

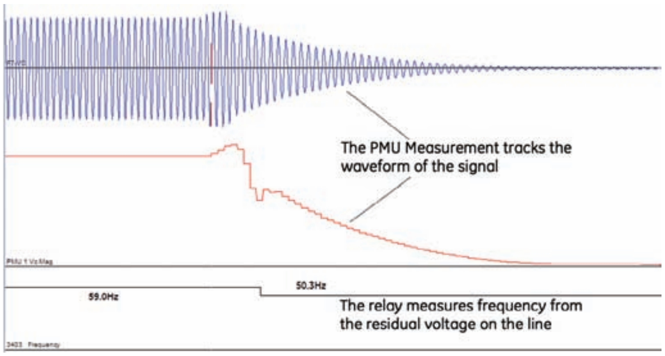

**Figure 12.**
*Phase C voltage decays after the breaker opens. The PMU measurement tracks the dynamic of this signal. The measured frequency registers the actual 50.3Hz resonant frequency between the line and its shunt reactors.*

The phase A voltage registers small values coupled via the shunt reactors after the breaker is opened and the fault removed. It is worth observing the phase angle of this voltage as measured via the synchrophasor algorithm. Figure 13 displays the phase A voltage angle. Before the fault the angle changes at the rate of 360deg/sec because it is reported at 60 times a second while the signal is of 59Hz ((60-59)*360deg/sec). When the voltage is driven by the 50.3Hz resonant fre-\quency on the disconnected line, the angle changes much faster at (60-50.3)*360deg/sec = 3500deg/sec, or one full revolution every in less than 100ms.

Examples presented in this section demonstrate the power of synchronized measurements to post-mortem analysis, including faults. Also, they depict secure co-existence of protection and PMU functions on the same IED platform.

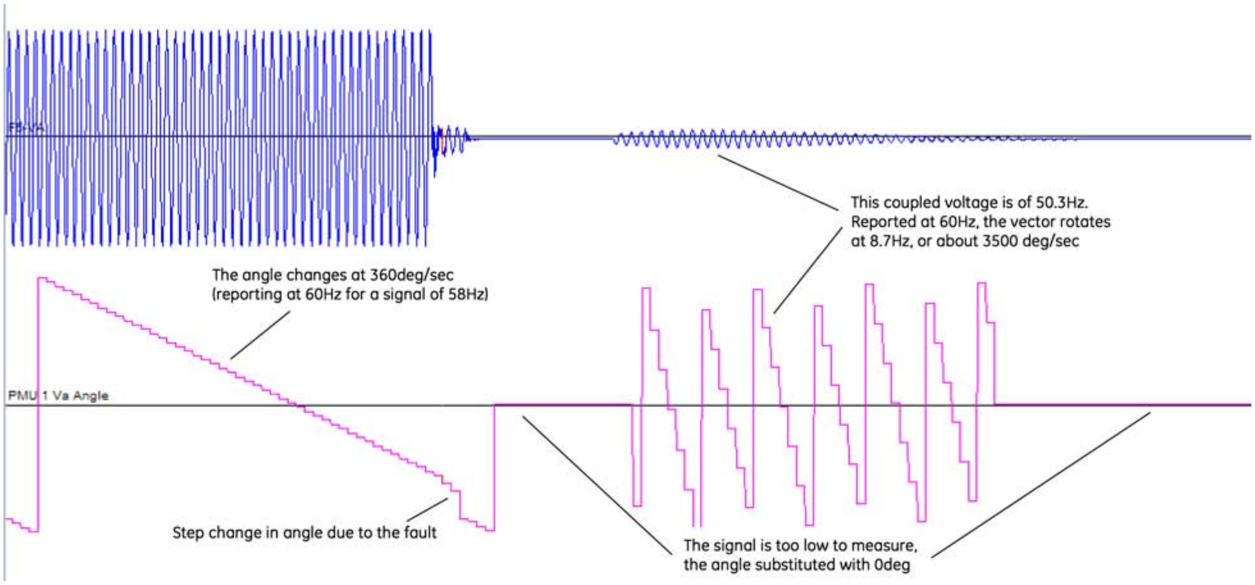

**Figure 13.**
*Phase A voltage coupled after the breaker opens. The PMU measurement reflects the frequency of this signal (seen as the rotating phase position of the voltage vector).*

## 13. Testing Recommendations for PMUs integrated with Protective Relays

Protection and control platforms integrating PMU functions should be tested in both protection and PMU modes of operation.

The protection functionality shall be tested given specific evaluation and approval philosophy for protection and control relays. During those tests the PMU functions should be enabled and configured in a way representative for a typical or worst-case future application. Similarly, the PMU functionality should be tested with a set of protection functions enabled and configured to reflect typical or worst-case future applications.

Having both sets of functions enabled and configured allows identifying any natural or unintended interactions between the two functionalities.

While the above general rules are followed, a few specific tests are worth recommending as follows:

- Speed of response of key protection function shall be checked during PMU-related activities. This includes normal PMU operation and extra activities such as coincidence of a system fault with a PMU command issued towards the IED from the PDC, local recording being initiated or in progress, retrieval of local records, and so on.

- Accuracy and integrity of key protection functions shall be checked during increased PMU activity.

- Accuracy and speed of response of key protection functions shall be checked during off-nominal frequencies. This includes steady state frequency deviations as well as frequency ramps. Modern protective relays are typically designed to retain full functionality under steady state off nominal frequencies, and exhibit only slightly degraded performance under frequency ramps, with the extent of degradation depending on the rate of frequency change. Increased demand on PMU accuracy under abnormal frequency conditions may result in shifting the design targets - potentially impacting performance of the core protection functions of the device.

- PMU functionality shall be checked under fault conditions. This includes any impact on accuracy after the fault is cleared, as well as integrity during the fault condition. For example, are all data frames produced during the fault or some of them may be lost? Is the post-fault steady state accuracy as expected or is the disturbance is having a long lasting impact on the accuracy of subsequent measurements?

- Integrity of both protection and PMU functions shall be checked under periods of simultaneous activity. For example, a command frame can be issued toward the IED just before a fault is applied – response to the fault should be checked as well as response to the command frame.

- Integrity of protection functions should be checked under impairments of IRIG-B input signal. Having to correlate measurements with the absolute time, IEDs implementing PMU functions may become affected by impairments of the IRIG-B timing signal. Adding noise, particularly to generate spurious 1pps patterns, or invalidating the time and date code is a meaningful check when overlaid on fault conditions. Step changes in time and date generated at the IRIG-B clock, or leap seconds, are good tests as well. Overall integrity of protection – both speed and selectivity – should be verified under such abnormal activities of the IRIG-B input.

- Communication impairments related to the PMU-PDC data exchange should be tested with respect to integrity of key protection functions. Classical channel impairments such as bit error rates corrupting the packets, multiple requests, invalid requests, etc. should be placed simultaneously with fault conditions. Selectivity and speed of protection should not be compromised.

- Dynamic response of the synchrophasor measurements under fault conditions should be tested and understood. The C37.118 standard does not mandate any specific performance under dynamic conditions, such as during system faults. However, PMU records will be a valuable source of information for post-mortem fault analysis. Response of a particular synchrophasor algorithm under fault condition needs to be tested and understood before the records can be used for fault analysis.

## 14. Synchrophasor Measurement Algorithm

This paper describes implementation of synchrophasors on modern relay platforms. Our particular implementation uses an optimized algorithm aimed at measurements under dynamic conditions.

Under steady-state operation of a power system at a constant known frequency, the appropriate definition of a synchrophasor is intuitive and obvious, and is the one specified by the C37.118 standard [3]. However, during dynamic conditions, it is not as clear what the definition should be. Also it is well known that off-nominal frequency operation [5] or power swings [6] can cause issues in the accuracy of the results of a classical phasor computation. For example, a power swing is actually equivalent to at least two closely spaced, distinct power frequencies with comparable amplitudes. Which one should be reported? Our implementation uses a multi-parameter model that resolves these issues, as well as matching the classical model under steady state operation at a single frequency.

Under steady-state conditions, a synchrophasor is the cosine and sine projections of a power system signal, at whatever frequency the power system is operating [3]. It is not necessary or likely for the power system to be operating exactly at the nominal frequency. The phase angle of a synchrophasor is defined to be the angle between the reporting time-tag and the peak of the signal, at the actual frequency [3], so the issue of steady state off-nominal frequency does not arise in the definition of synchrophasors, only in their implementation [5].

The question arises how to define a synchrophasor during changing conditions? A logical approach is to define the power system signals to be projections of phasors that themselves are changing in time:

$$x(t) \approx \sqrt{2} \cdot Real\left(\overline{\mathbf{X}}(t) \cdot e^{j2\pi \cdot f \cdot t}\right)$$

$$x(t) = \text{instantaneous current or voltage} \qquad (1)$$

$$t = \text{time}$$

$$\overline{\mathbf{X}}(t) = \text{time varying phasor}$$

Definition (1) encompasses both changes in phase angle as well as changes in amplitude, so it models both the off-nominal frequency case, as well as power swings. The value of a phasor at a time-tag is simply the value of the time varying phasor in (1) when the time is equal to the value of the time-tag of the reported synchrophasor.

It is well known that the classical algorithms for computing phasors on a per-phase basis from sequences of samples incur errors during off-nominal frequency operation [5], [7] or during power swings [6]. Most of the errors cancel out in positive sequence phasors that are computed from per-phase phasors, provided that the negative sequence value is equal to zero. If there is some negative sequence, the errors in per-phase phasors do not exactly cancel, so there is residual error in the positive sequence phasor.

It is impossible in principle to tell the difference between off-nominal frequency operation and a constant time rate of change of the phase angle of the phasor. In either case, if the sampling rate is not matched to the power system frequency, errors arise [5], [7] in the classical algorithms. For constant amplitude and phase angle signals, the computed per-phase phasors trace an elliptical trajectory [7]. The eccentricity of the ellipse can be predicted from the frequency. If the frequency is known, the errors in the per-phase phasors can be exactly compensated, though there will still be an issue of incomplete harmonic rejection. Two other solutions to the off-nominal issue include frequency tracking and re-sampling.

The off-nominal frequency effect is equivalent to a backward rotating error [6]. If the underlying phase signals are balanced, the backward rotating errors cancel in the positive and zero sequence phasors computed from phase values, although there will be an apparent negative sequence component. If the phase signals are not balanced, there is trouble in general.

The power swing case has been analyzed in [6], and one method for greatly improving the accuracy using a raised cosine windowing function has been described.

Another method, described here, uses a Taylor's series expansion to represent a time varying phasor to address both the off-nominal frequency effects as well as power swing issues with a simple extension of the classical algorithms for computing phasors.

To solve this problem our implementation assumes both the magnitude and "phase" of a phasor to be linear function of time, and estimates such varying phasors to fit them best to the measured waveforms. As a result our model gives much better response under dynamic system conditions.

## 15. Conclusions

This paper discusses various implementation issues related to integration of synchrophasor measurements and PMU functionalities on microprocessor-based relay platforms.

The paper alerts prospective users to possible pitfalls of the integration and allows making a more informed decision based on the understanding of both the synchrophasor and relay technologies.

The paper discusses sample tests that could be used to probe the robustness of the integrated PMU/relay implementation.

We presented one particular way of implementing synchrophasors that calls for practically no changes to the underlying relay architecture. The sampling, frequency tracking, data collection and manipulation processes have been preserved with no changes. All the synchrophasor related calculations and operations are kept completely separated within a framework of object-oriented programming. The only change is the addition of one new interrupt between the CPU and DSP. This interrupt is not used at all by any of the protection functions, thus minimizing any danger of unintended changes.

The presented implementation is based on a novel multi-parameter algorithm for estimating synchrophasors under dynamic system conditions. The approach assumes slow transients in the estimated phasors and solves the assumed multi-parameter signal model accordingly to provide for both accurate and fast synchrophasor measurements.

Test and simulation results prove equivalency with classical algorithms under steady states, and superior performance under system transients.

Test results on the actual hardware allow claiming accuracy of approximately twice as good as the most stringent requirements of the IEEE Std. C37.118.

It is justified to assume that synchrophasors will follow SOEs, DFRs, RTU and metering functions and become universally integrated on modern relay platforms. This is not only possible with future new platforms, but also within existing presently used relays. Careful engineering allows safe implementations and the accuracy equal if not better than standalone PMUs.

Integrated PMUs will allow wider penetration of this new technology, faster learning curve, and cost savings related to purchasing, installing and operating the equipment.

## References

[1] A.G.Phadke, J.S.Thorp, M.Adamiak, "A New Measurement Technique for Tracking Volt-age Phasors, Local System Frequency, and Rate of Change of Frequency," IEEE Transactions on PAS, May 1983.

[2] A.G.Phadke, "Synchronized Phasor Measurements," IEEE Computer Applications in Power, April 1993.

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

[4] D60 Line Distance Protection System, Instruction Manual, GE Publication GEK-113270. Available at www.multilin.com.

[5] T.Funaki, S.Tanaka, "Error Estimation and Correction of DFT in Synchronized Phasor Measurement," Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES Volume 1, 6-10 October 2002, pages: 448-453.

[6] J. A. de la O Serna, K.E.Martin, "Improving Phasor Measurements Under Power System Oscillations," IEEE Transactions on Power Systems, February 2003.

[7] M.Adamiak, B.Kasztenny, W.Premerlani, "Synchrophasors: Definition, Measurement, and Application", Proceedings of the 59th Annual Georgia Tech Protective Relaying, Atlanta, GA, April 27-29, 2005.