Real time synchronized data consumption and archival for the power grid

March 13-14. 2012 Pittsburgh, PA

C. H. Wells and M. Heere, OSIsoft, LLC

Abstract:

Secure wide area transport of high speed time synchronized data with minimal latency to a centralized location is discussed. Near real time consumption of these data by multiple clients and the types of basic calculations are outlined, with code examples of unwrapping of discontinuous phase angle data. Archival volumes and data compression are reviewed, and compression examples with real PMU data are shown. This will debunk the "myth" that PMU data must be stored in uncompressed form.

Introduction

One of the innovations in the power industry is the growing acceptance of time synchronized data. These data can be used for improved situational awareness, wide area protection and control. The rapid adoption is in part due to the $400 million Smart Grid Investment Grant program funded under the ARRA Act of 2009. This is program will result in the installation of over 1000 PMU across the USA. The WECC specifically will have over 300 PMUs sending data to its central server by the end of 2012.

One of the innovations is the measurement of the absolute phase angle for both current and voltage for each phase. The measurements are made simultaneously based on GPS clocks that provide time accuracy to better than one microsecond. Additionally, the measurement devices (called PMUs), are required to measure at specific rates with all sampling starting at the top of second. The standard defining the measurements is known as IEEE C37.118 (2005). A new standard known as IEEE C37.118.1 and .2 will be published in 2012. One of the innovations in the measurement system is that the measurement must be accurate to within one percent total vector error. This requires time measurement accuracy better than ± 26 μ seconds to obtain a one percent TVE (total vector error), see Figure 1 below for a definition of TVE. This is defined in the C37.118 standard. One of the basic issues with most PMUs on the market today, is obtaining an accurate time signal as input to the PMU. There is not a general standard on how PMU manufacturers obtain accurate time sources. The most common one used is called IRIG-B; however, the interfaces to the PMUs are standardized to the level where any PMU can accept any IRIG-B signal. This is well documented in the PMU literature available at the NASPI.org web site.

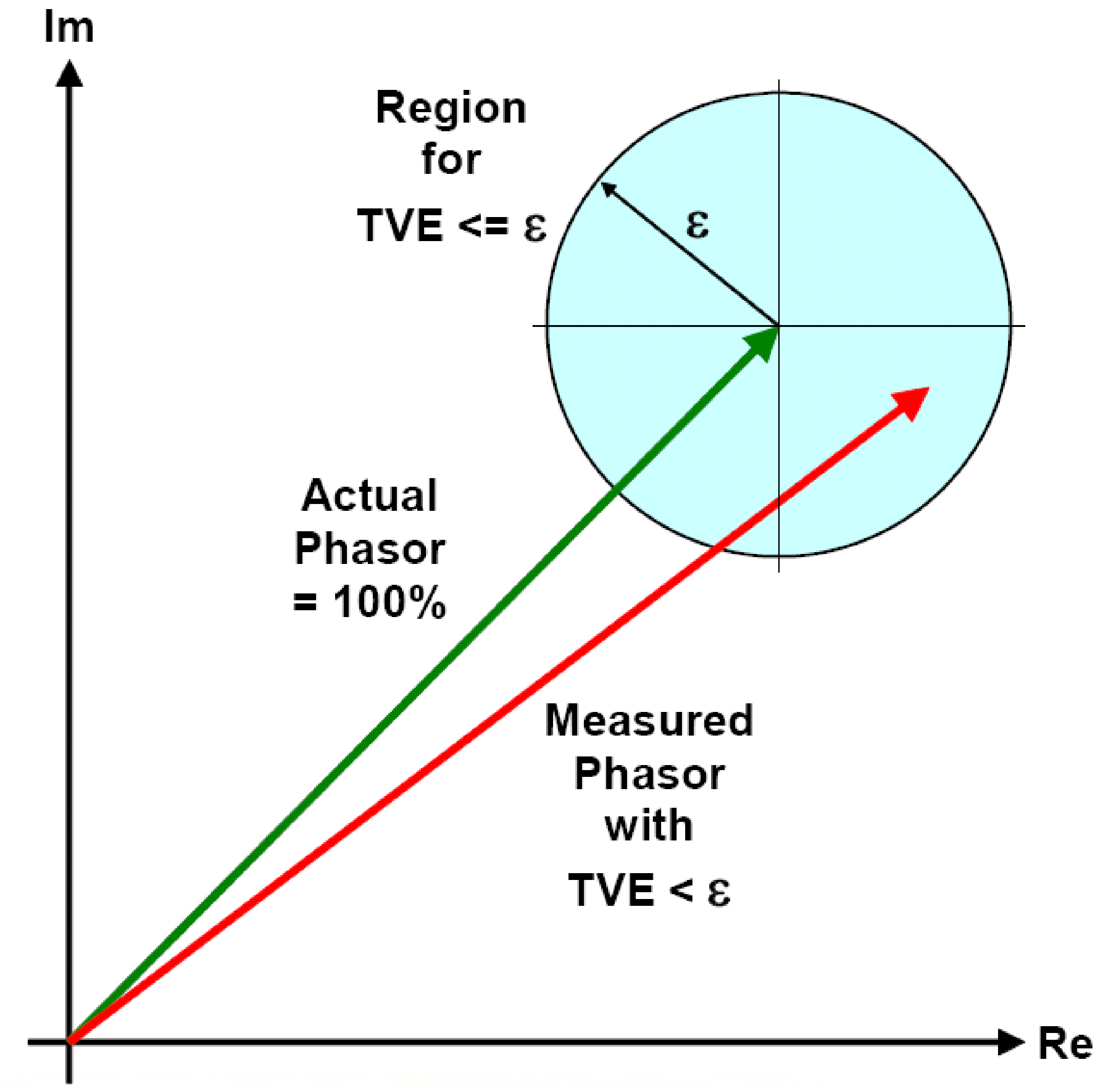


Figure 1 Definition of Total Vector Error

Another issue is the source of the signal going into the PMUs. This is often from protection grade CT and PTs. These normally operate at or near the very bottom of their range in the non-linear region of the CT or PT. So it is not uncommon to have a low voltage error of up to 7 percent from the CT/PT. This is often overlooked by utility companies since they frequently do not see the value of replacing the protection grade CT and PT with measurement grade units. Although some PMUs have the ability to compensate for the non-linearity, few installations use the calibration tools to increase the accuracy of the overall system since the circuit has to be de-energized to calibrate the devices.

The typical measurement accuracy of a PMU meeting IEEE C37.118 standard exceeds 0.1 percent error. This includes frequency and rate of change of frequency (ROCOF).

The standard reporting rates are 10, 12, 15, 20, 25 and 30 Hz; however, most PMUs being installed today are reporting data at 60 Hz rates and some in China are reporting at 100 Hz. There is doubt that certain measurements retain their accuracy at the higher rates, specifically frequency and ROCOF.

The PMU is connected to the low voltage side of the CT and PT connections for each phase. These are typically less than 480 volt RMS and less than 10 amp; however, most of the PMUs on the market today only support signals up to 1 amp. This is to reduce the cost of their instrument. This increases the cost to the utility company and lowers the overall accuracy of the current measurement.

The IEEE C37.118 interface requires time accuracy be reported in each message. The data include indications of a pending leap second insertion, seconds after a leap second insertion, and the current quality of the time stamp. Data quality is also included; however, this signal is either good or bad.

The low voltage signals connected to the PMU are typically sampled at high precision at high data rates, often in the kilohertz range. The sampling is done with ADC converters of the manufactures choice and most use 12 to 16 bit ADC hardware and software. However, most PMU manufactures do not sample synchronously with the GPS top of second signals. However, at least one PMU manufacturer (Arbiter 1133a), performs synchronous sampling and ADC conversion.

The method of computing the absolute angle is left to the PMU manufacturer; however, they are required to meet the IEEE C37.118 TVE accuracy requirements. Most manufactures use the Phadke method of computing angles. However, this method is known to be dependent on the actual frequency of the system. That is as the frequency drifts from the fundamental (60 Hz in the USA), the accuracy of this method deteriorates. The accuracy drift with frequency is substantial (up to 5 percent for one Hz offset) without compensation. Most vendors indicate some compensation method. However, one manufacturer does not use Phadke method, and meets the TVE requirement across ± 10 Hz range (Arbiter 1133a).

The most common PMUs report voltage and frequency phasors, most often in polar coordinates. However, for vector computation, it is often far faster to perform the computation in rectangular coordinates. For example computation of real, reactive, and apparent power is far more efficient in rectilinear coordinate system. Additionally, the phase angle data is discontinuous. It normally wraps around $\pm \pi$ radians (normally reported in 180 degrees); however, some of the older PMUs and those from FNET wrap between zero and 2π radians and the angle is reported in radians.

Messages emitted from the PMU contain a header, timestamp and data quality, frequency, ROCOF, phasor data, followed by analog data. The structure of the data components in the message are contained in a special message called the CONFIG2 block. This message can be requested by the data receiver, or can be broadcasted by the PMU at regular intervals. This is the only method to determine what each PMU is reporting. The message can be transmitted using UDP or TCP formatted messages using IP protocol. Most companies are using TCP for the CONFIG2 and START commands, and UDP for data. Recently, it has been discovered that UDP buffer overruns reported by some companies can be eliminated by reconfiguring the UDP input buffer size (Sisco, paper at NASPI Orlando-2012). This must be done via program control, since the buffer sizes cannot be configured manually in either Windows or Linux operating systems.

Conversion messages to data streams

Each message contains four or more data values (frequency, ROCOF, voltage magnitude and angle), but typically messages will contain 20 to 80 data values in multiple different binary formats. These messages have to be unpacked and converted to values that can be used by software applications. Typically this is done by an application software package collecting the data from the PMU or from a PDC (Phasor Data Concentrator). A PDC reads data from multiple PMUs and merges data at the same time stamp into a single message containing a single time stamp, but with time quality for each PMU. PDCs can send the same data to multiple software clients. This may often be needed since most PMUs cannot multicast their data to more than several clients.

The PDCs add latency to the messages since it has to wait a specified period of time to assure that all messages with the same time stamp arrive at the PDC. The combined message is then re-emitted to multiple clients. This delay is normally set to about 50 milliseconds. This added latency builds as additional PDCs are added in the network. Typically two or more PDCs are used to provide data to a central location. See Figure 2 below:

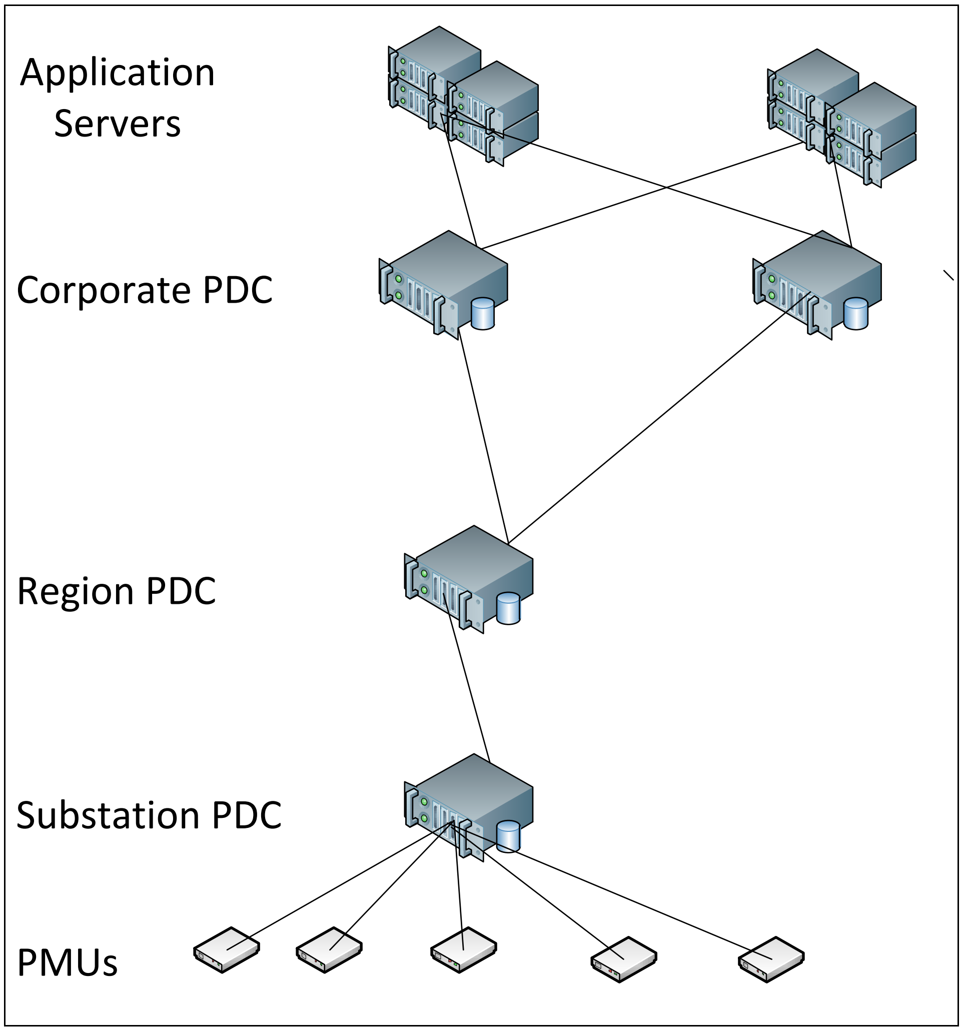


Figure 2 PDC Stacking

Client applications

The client reading the data stream must request the CONFIG2 block (or be knowledgeable of its structure) in order to convert the IEEE C37.118 messages to data streams with time stamps, and data quality. One common application directly reading C37.118 messages the RTDMS product from EPG. This is a product supported by the Department of Energy and is free to utility companies.

The PI System reads C37.118 messages and converts these into separate data streams. It does this using a standard OSIsoft, LLC. interface software package. This can be loaded onto any Windows machine that has read and write access to the IP address of the PMU and a PI server. The architecture for this is shown in Figure 3 below:

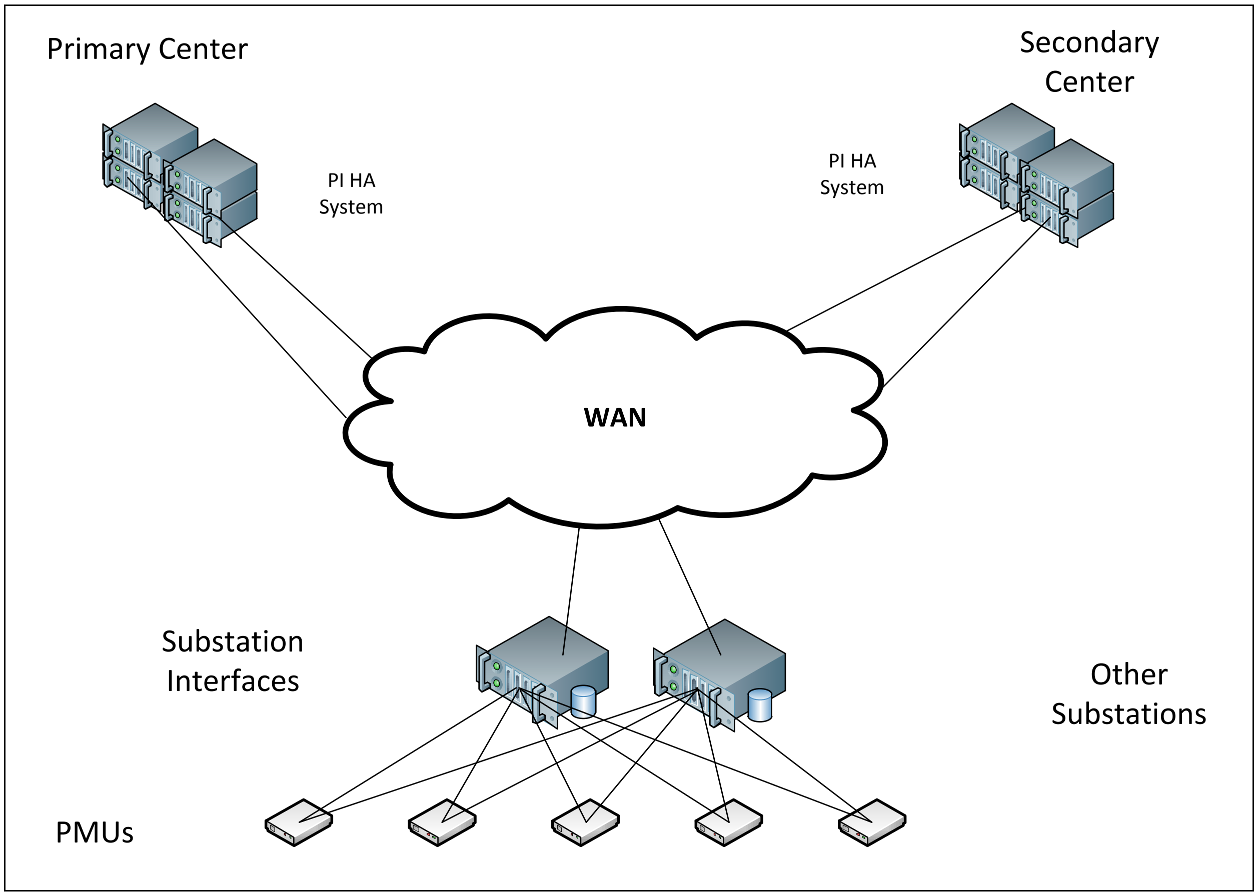


Figure 3 PI Interface architecture

The interface also provides options to send both polar and rectangular coordinates to the PI server. This can improve the speed of computations on the vector data; for example calculation of the real, reactive and apparent power from voltage and current phasors. The interface parses the C37.118 message into data streams, each having a time stamp, value and quality. Additionally, the PI interface includes two additional tags: EVENTSEQ and COMPQUAL. The EVENTSEQ is an integer representing the event order starting at the top of the second. Normally this will range from 0 to 59 for 60 Hz data. The COMPQUAL tag is the OR of bad data (STAT word) and any of the thirteen bad time states (FRACSEC word) that are defined in the IEEE C37.118 standard. This provides the application with the means to rapidly determine the overall quality of the data. We suspect that most of the time the COMPQUAL will be zero (= good time and good data).

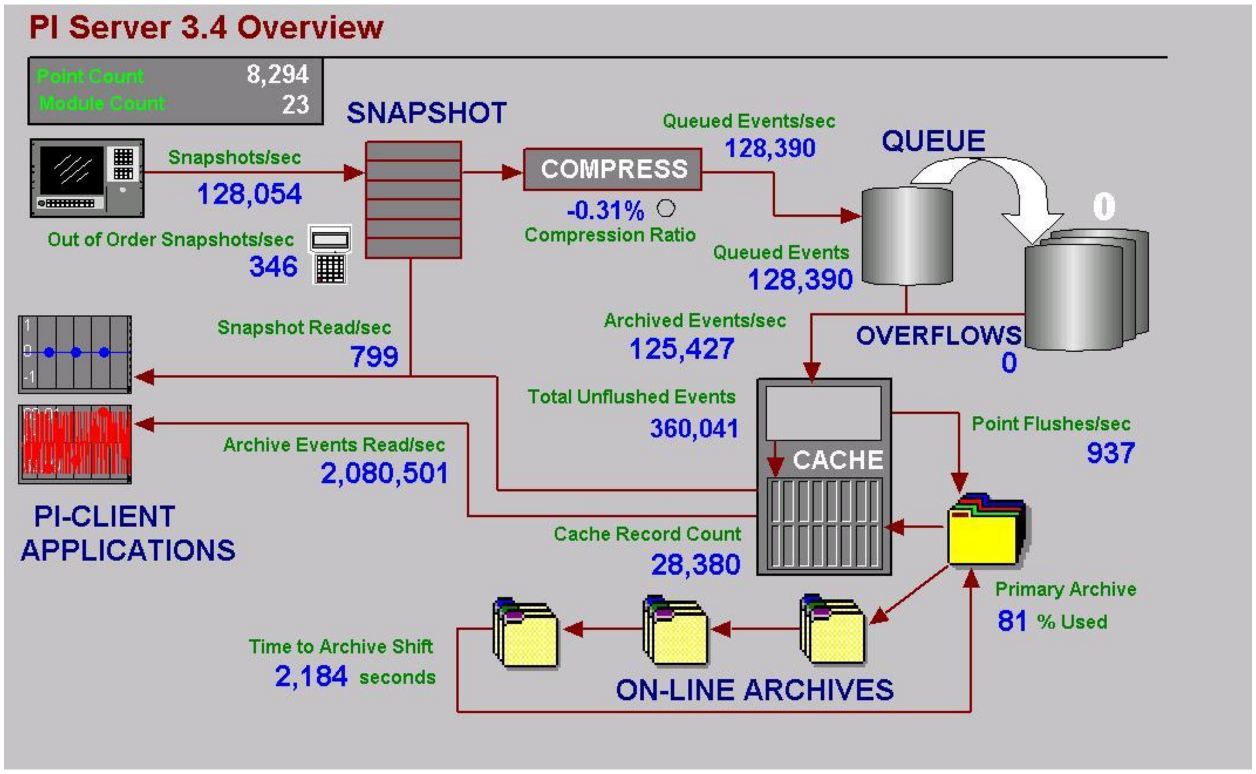
The Snapshot

The tags created by the PI C37.118 interface are sent via an unpublished binary protocol to any number of PI servers via buffers located in the interface. The messages may be sent with high levels of encryption. The buffers are needed when communications to the PI servers fail. In these cases the buffer is unloaded in chronological order at much faster than real time speeds (typically 100 to 1000 times faster). The snapshot is memory resident in the PI server. Each tag value and its status are stored in memory. These values are available to hundreds of clients in real time. The PI system design is to provide archived data to hundreds of clients hundreds of time faster than the data arrives. A typical large PI system may have several thousand client reading data simultaneously. Note that an achived data point in the PI system might be only 16 mS old.

Performance of the PI System

The current PI system has been extensively tested using simulated data. The following tests were performed using data from a PDC simulator developed for WECC performance testing.

The results are shown below:



This shows 128k events going into to the PI server, no compression is preformed. The system is archiving 125k events per second and delivering 2 million events per second to clients.

Compression
How PMU data gets into the PI system

The PI C37.118 interface (software) receives data at fixed rates defined in the IEEE C37.118 specifications at: 10, 12, 15, 20, 25 and 30 samples per second. This software can run on any windows machine that can read and write to the IP address of the PMU. Most PMUs can report data at 60 Hz and most current users are planning on acquiring data this higher rate.

Exception handling

The interface reads messages sent from the PMU, evaluates each measurement in each message to determine if there are significant events; if so, it reports the new events to the PI Server. This process is called exception reporting. The current value is compared to the previously sent value and, if it is different; the value, time stamp, and data quality will be sent to the PI Server.

The new value is not reported unless: the difference between the new value and the last value is greater than the exception deviation specification and the difference between the times of the new and the last value is greater than or equal to the exception minimum time specification: or, the difference between the timestamp of the new value and the timestamp of the last reported value is greater than or equal to the exception maximum time specification.

ExcDev

The ExcDev attribute (Exception Deviation) specifies in engineering units how much a value may differ from the previous value before it is considered to be a significant value. The ExcDevPercent attribute specifies the same thing as a percentage of the Span attribute. A typical value for phasor angle would be 0.1 degree. The typical reported accuracy of a PMU measurement angle is 0.1 degree. So if the new angle measurement is less than 0.1 degree from the previous angle, no exception is reported.

ExcMin

The ExcMin attribute (Exception Minimum) is the time delay after previous value was collected. This is used to suppress noise. It is specified in seconds. A new data value that is received before the end of the ExcMin interval will be discarded. For example if the data collection rate is 1/60th of a second and the ExcMin is 1/60th of a second and if a new value arrives at 1/120th of a second after the previous value, it will be discarded.

ExcMax

The ExcMax attribute (Exception Maximum) puts a limit on the length of time that values can be discarded due to exception. For example, it is possible for the incoming data to be a single value for many days. If ExcMax is set to 60 seconds (one minute) then a value will be stored if the previous event timestamp was more than 60 seconds before that.

Scan Attribute

If Scan is OFF, the interface will not read the PMU and therefore no events will be sent to the PI System.

Snapshot

A new event entering the PI System from an interface is sent to the Snapshot Subsystem. The snapshot is the most recent value for a point. It can be viewed as a one deep memory resident buffer for the incoming data. When a new event comes in, it becomes the new snapshot. The previous snapshot is evaluated according to the compression specifications and is either sent to the Event Queue or discarded.

Any event that has a timestamp older than the snapshot is considered a significant and will be put directly into the Event Queue of the Archive Subsystem.

Point values are always stored in full precision (defined in the interface, this can be a Float 32 or Float 16, or INT 16 and in some cases Float 64) in the Snapshot. Scaling, if applicable, is applied when the event is stored into the Archive.

Compression

When a new Snapshot arrives, the previous one is evaluated according to the compression specifications. If it exceeds the compression specifications, it is sent to the Event Queue. If not, it is discarded.

There are three instances where an event will bypass compression and be put in the Event Queue:

- If the Compressing attribute for the point is set to OFF.
- If the timestamp is older than the timestamp of the current snapshot. Such an event is considered ‘out of order.’
- If the Status attribute of the Point has changed.

The compression method allows orders of magnitude more data to be stored online compared to other conventional systems. The compression method is called ‘swinging door compression.’ Swinging door compression discards values that fall on a line connecting values that are recorded in the Archive. When a new value is received by the Snapshot Subsystem, the previous value is recorded only if any of the values since the last recorded value do not fall within the compression deviation “blanket”. The deviation blanket is a parallelogram extending between the last recorded value and the new value with a width equal to twice the compression deviation specification.

Each tag has three attributes that comprise the compression specifications: CompDev (compression deviation), CompMin (compression minimum time), and CompMax (compression maximum time).

CompDev is the half-width of the deviation blanket (as shown in the illustration). CompDevPercent is similar to CompDev, but it specifies the compression deviation in percent of Span rather than in engineering units.

The compression specifications work in a similar way to the exception specifications. Just like exception reporting, compression is a filter. The difference is that the exception specifications determine which events should be sent to PI, whereas the compression specifications determine which of the events sent to PI should go into the Archive.

CompMin and CompMax are limits that refer to the time between events in the Archive. A new event is not recorded if the time since the last recorded event is less than the compression minimum time for the point. A new event is always recorded if the time since the last recorded event is greater than or equal to the compression maximum time.

The maximum time specification does not guarantee that a value will be written to the Archive within a certain time. The Archive waits for events to be sent to it. It does not check to see if a point has timed out. It does not 'create' new values. If a value exceeds the Compression Deviation Specification it will be archived. A compression deviation blanket drawn to this point would not include all points since the most recently archived value, so the previous value would be archived. The compression parameters can be changed to produce efficient archive storage without losing significant process data. The compression maximum time is usually set to one value for all points of a given type. It should be large enough that a point that does not change at all uses very little archive space. A compression maximum time of one second is a good choice for phasor data.

Use the compression minimum time (CompMin) to prevent an extremely noisy point from using a large amount of archive space. This parameter should be set to zero for any point coming from an interface that does exception reporting. In this case, the exception minimum time should be used to control particularly noisy points.

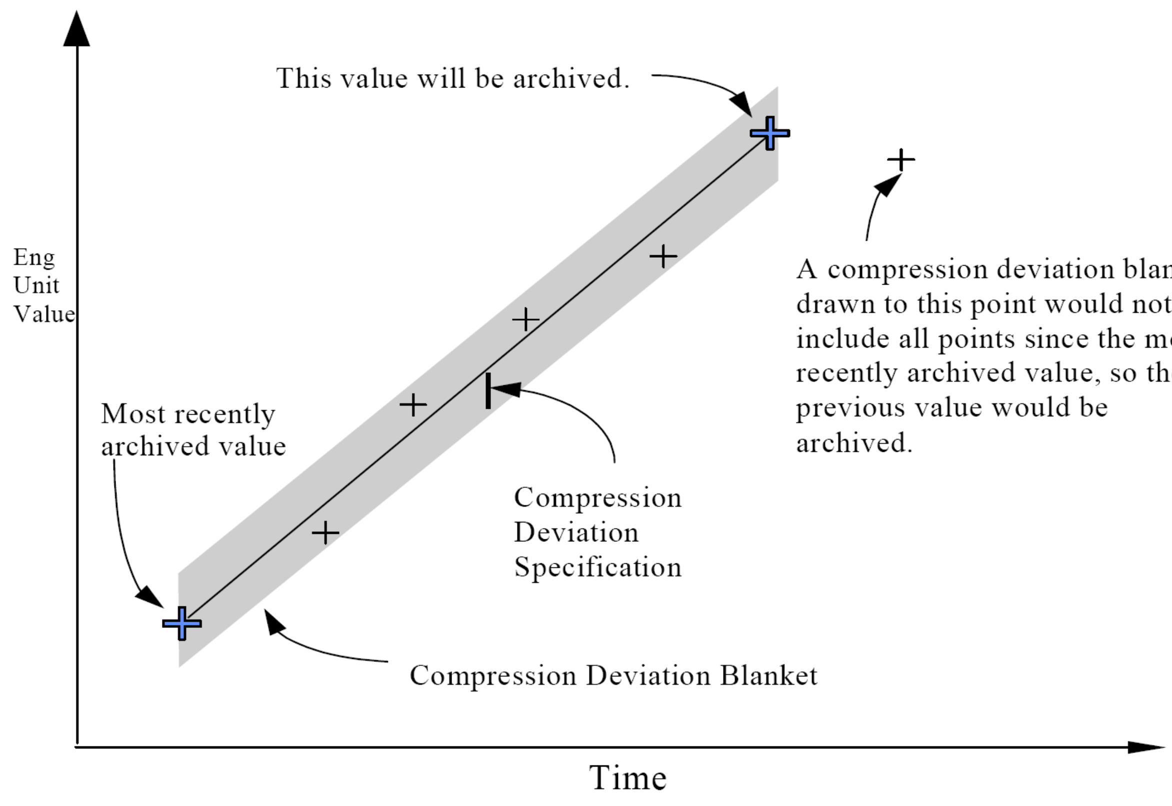

This value will be archived.
Eng Unit Value
A compression deviation blanket drawn to this point would not include all points since the most recently archived value, so the previous value would be archived.
Most recently archived value
Compression Deviation Specification
Compression Deviation Blanket
Time

Figure 4. Definition of the compression deviation blanket (parallelogram)

The most significant compression parameter is the deviation specification, CompDev. This parameter is often adjusted after the point is defined. A reasonable starting point is one percent of span. Another way to look at CompDev is to consider it a rate of change. For example, for a 60 Hz sample rate, how fast does the process normally change in one sixth of a second. The goal is to filter out instrument noise and still record significant process changes. The effect of changing the compression deviation is not predictable.

For digital points, any change is a significant change. Only the compression maximum and minimum time are important. The compression deviation specification is ignored for digital points.

Step Flag

The step attribute setting affects both display and compression.

Data for points with this attribute set to 1 is assumed to remain fixed between events, whereas for points with step=0 data is assumed to change linearly between valid numeric events.

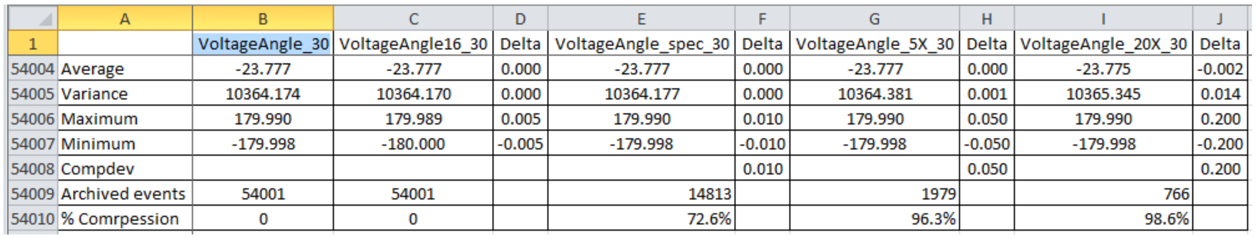
The swinging-door compression, explained above, is not used when the step flag is set. Instead, an exception calculation is applied using the CompDev value. If the absolute difference between the current snapshot and the last archive value is greater than CompDev then the snapshot is sent to the archive.

A more detailed explanation of compression may be found in the Appendix.

Results of Compression...

Data files were created for each of the following measurements: Phase A voltage angle (raw discontinuous angle data, Phase A Voltage magnitude, Phase A current angle, Phase A voltage angle, Phase A current magnitude, Positive Sequence VAR, Positive Sequence VA, and frequency. Three compression settings were tested: (1) instrument stated accuracy, (2) five times this value, and (3) 20 times this value. Here is a link to the compression testing data. Applications using phasor data should try running their applications on these data to determine specifically if the decompressed data affects their applications. Based on our analysis of real data, we see no loss of information.

Voltage Angle results



	A	B	C	D	E	F	G	H	I	J
1		VoltageAngle_30	VoltageAngle16_30	Delta	VoltageAngle_spec_30	Delta	VoltageAngle_5X_30	Delta	VoltageAngle_20X_30	Delta
54004	Average	-23.777	-23.777	0.000	-23.777	0.000	-23.777	0.000	-23.775	-0.002
54005	Variance	10364.174	10364.170	0.000	10364.177	0.000	10364.381	0.001	10365.345	0.014
54006	Maximum	179.990	179.989	0.005	179.990	0.010	179.990	0.050	179.990	0.200
54007	Minimum	-179.998	-180.000	-0.005	-179.998	-0.010	-179.998	-0.050	-179.998	-0.200
54008	Compdev					0.010		0.050		0.200
54009	Archived events	54001	54001		14813		1979		766	
54010	% Comrpession	0	0		72.6%		96.3%		98.6%	

Table 1 Comparison of three levels of compression

A total of 54001 samples were collected. The compression settings are shown in engineering units; in this case it is the precision of the instrument in angle (degrees). So directly from the Arbiter manual, the angle accuracy is stated to be 0.01 degrees. So at five times the specification, the compdev= 0.05 and at 20 times, the compdev = 0.20. The interesting observation is that at simply the accuracy setting of compdev, the compression results in 72.6 percent reduction in disk space.

Shown in Figure 5 is a trend plot of the uncompressed data and the "decompressed" data. This shows five traces of voltage angle as it transitions from -180 to +180 degrees. This might be the worst case for the compression algorithm. But clearly there is no discernable difference between two uncompressed curves and the three decompressed curves. So our conclusion is that decompressed data is valid for current known applications.

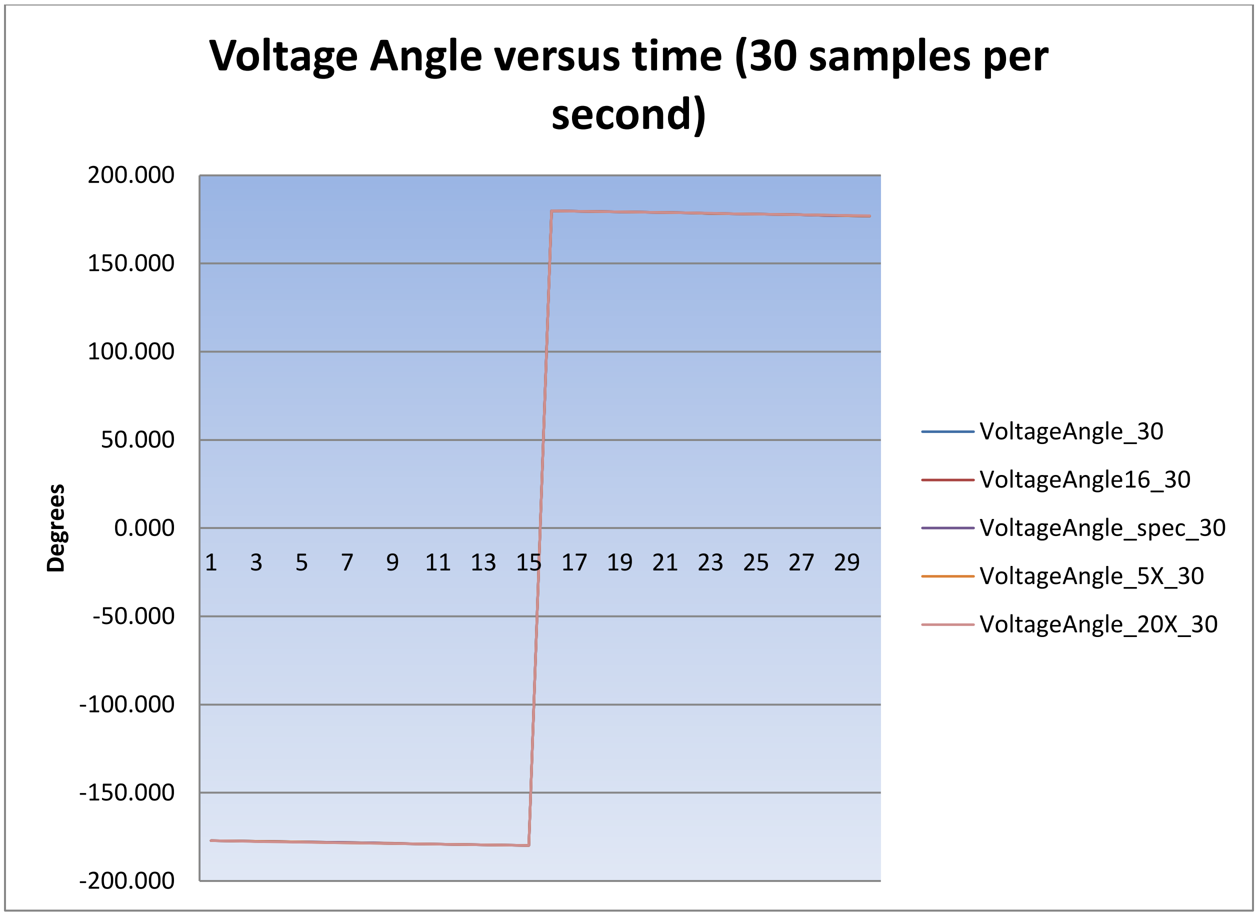


Figure 5 Trend plot of voltage angle as it transitions a discontinuity.

A second way to look at possible errors in decompressed data is to use an X-Y plot and determine the correlations between the "uncompressed" and the decompressed data under different compression settings.

An example is shown below in Figure 6.

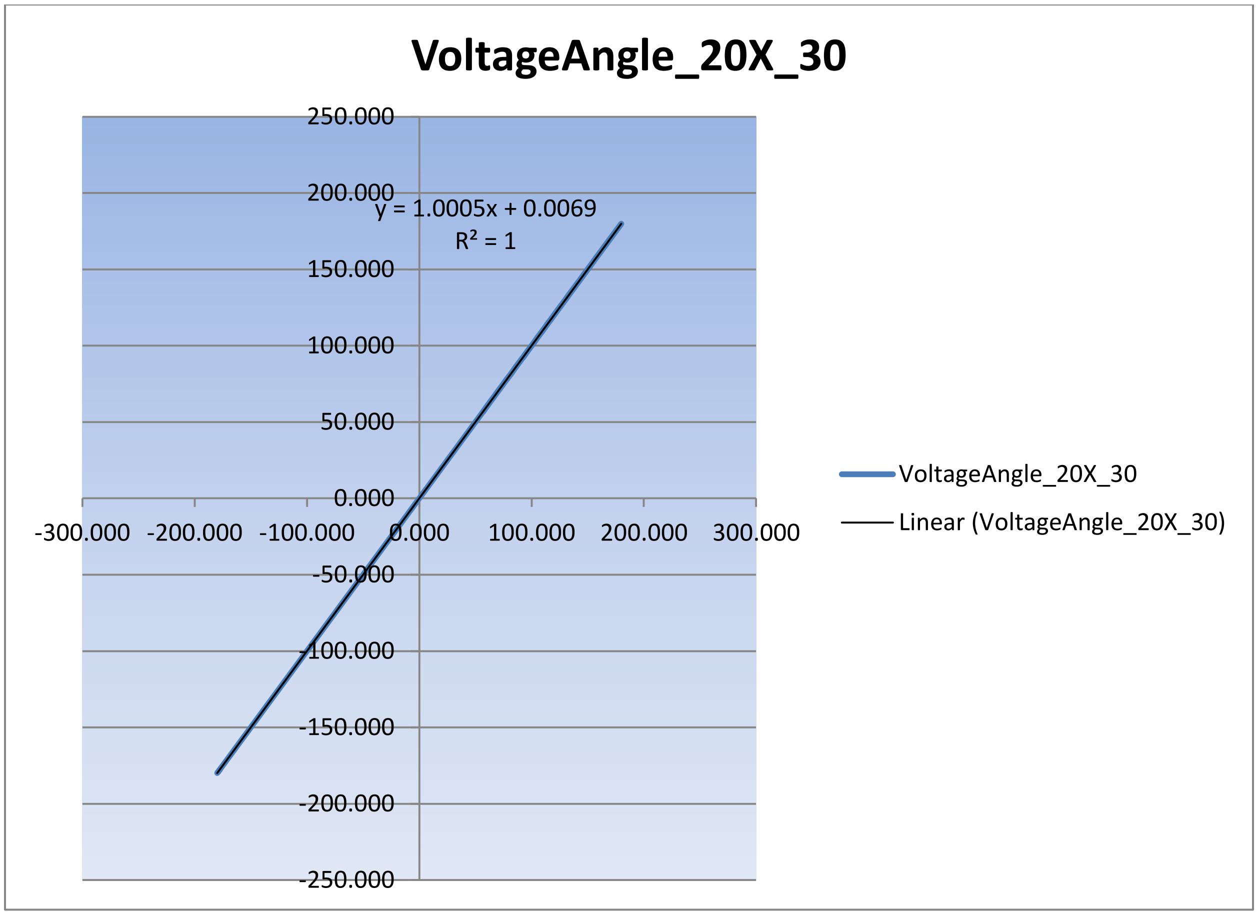


Figure 6, X-Y plot of uncompressed voltage data versus 20* compdev decompressed data

This is the cross correlation of the first 2500 data points in the data array. The correlation coefficient is very good, showing that the two data streams are essentially the same even at a compdev setting of 20 times the instrument specifications. The curve is essentially the same for any pairs of data in the set (there are 54001 rows in the data set).

A third way to look at the effects of compression on the data is to plot the trend lines as the angle crosses zero. This is shown in Figure 7.

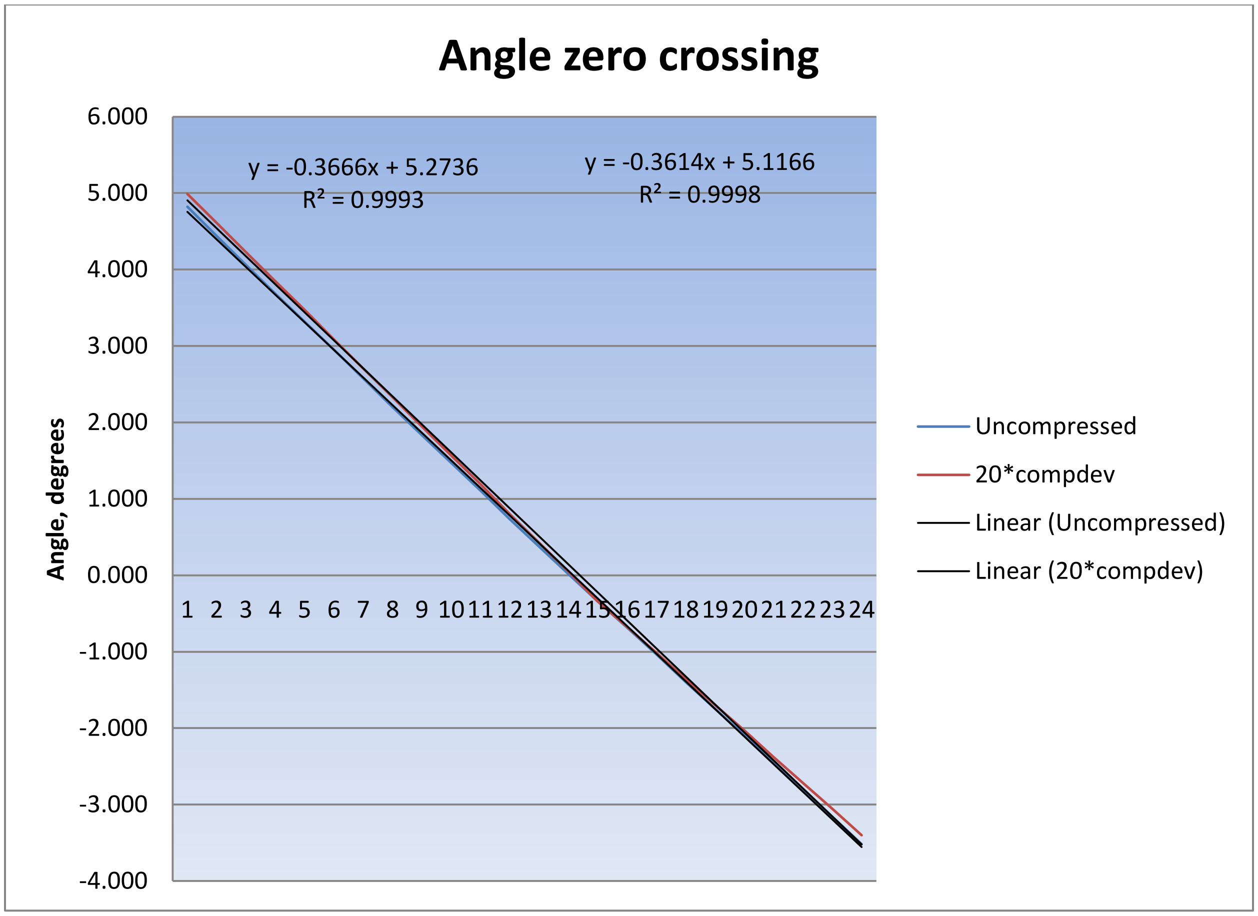
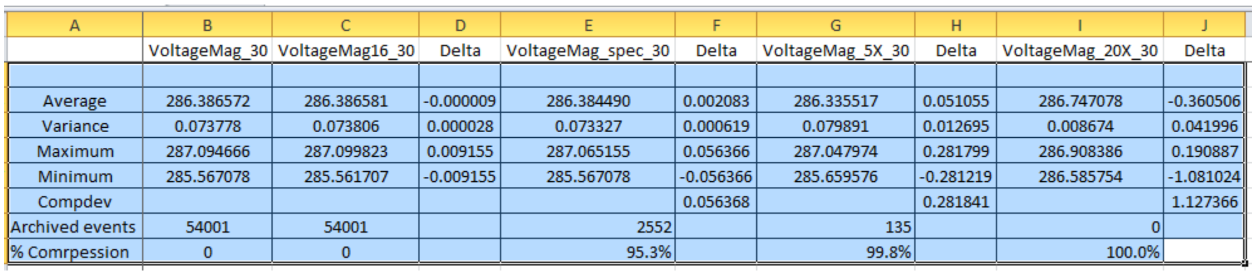


Figure 7. Angle zero crossing (correlation coefficient comparison of trend lines

Note that the correlation coefficients between the trend line linear fit are quite good as indicated by the R squared values of 0.9993 and 0.9998.

The following tables show the results of the compression testing.

Voltage Magnitude

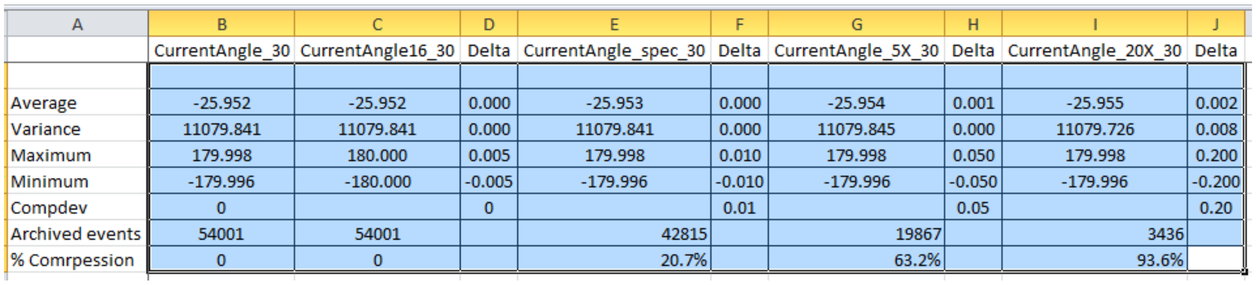


A	B	C	D	E	F	G	H	I	J
	VoltageMag_30	VoltageMag16_30	Delta	VoltageMag_spec_30	Delta	VoltageMag_5X_30	Delta	VoltageMag_20X_30	Delta
Average	286.386572	286.386581	-0.000009	286.384490	0.002083	286.335517	0.051055	286.747078	-0.360506
Variance	0.073778	0.073806	0.000028	0.073327	0.000619	0.079891	0.012695	0.008674	0.041996
Maximum	287.094666	287.099823	0.009155	287.065155	0.056366	287.047974	0.281799	286.908386	0.190887
Minimum	285.567078	285.561707	-0.009155	285.567078	-0.056366	285.659576	-0.281219	286.585754	-1.081024
Compdev					0.056368		0.281841		1.127366
Archived events	54001	54001		2552		135		0	
% Comrpession	0	0		95.3%		99.8%		100.0%	

Table 2 Voltage Magnitude comparison

One might wonder why there is 100 percent compression on voltage magnitude. This is because the voltage did not vary outside of the compdev specification while the data was being collected.

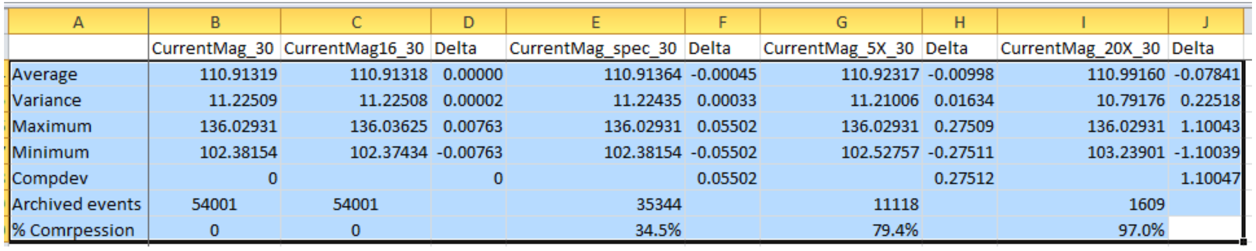
Current Angle



A	B	C	D	E	F	G	H	I	J
	CurrentAngle_30	CurrentAngle16_30	Delta	CurrentAngle_spec_30	Delta	CurrentAngle_5X_30	Delta	CurrentAngle_20X_30	Delta
Average	-25.952	-25.952	0.000	-25.953	0.000	-25.954	0.001	-25.955	0.002
Variance	11079.841	11079.841	0.000	11079.841	0.000	11079.845	0.000	11079.726	0.008
Maximum	179.998	180.000	0.005	179.998	0.010	179.998	0.050	179.998	0.200
Minimum	-179.996	-180.000	-0.005	-179.996	-0.010	-179.996	-0.050	-179.996	-0.200
Compdev	0		0		0.01		0.05		0.20
Archived events	54001	54001		42815		19867		3436	
% Comrpession	0	0		20.7%		63.2%		93.6%	

Table 3 Current Angle comparison

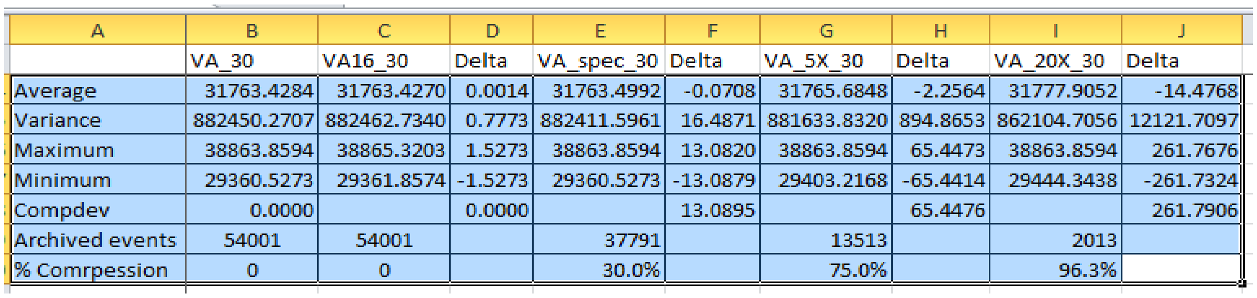
Current Magnitude



A	B	C	D	E	F	G	H	I	J
	CurrentMag_30	CurrentMag16_30	Delta	CurrentMag_spec_30	Delta	CurrentMag_5X_30	Delta	CurrentMag_20X_30	Delta
Average	110.91319	110.91318	0.00000	110.91364	-0.00045	110.92317	-0.00998	110.99160	-0.07841
Variance	11.22509	11.22508	0.00002	11.22435	0.00033	11.21006	0.01634	10.79176	0.22518
Maximum	136.02931	136.03625	0.00763	136.02931	0.05502	136.02931	0.27509	136.02931	1.10043
Minimum	102.38154	102.37434	-0.00763	102.38154	-0.05502	102.52757	-0.27511	103.23901	-1.10039
Compdev	0		0		0.05502		0.27512		1.10047
Archived events	54001	54001		35344		11118		1609	
% Comrpession	0	0		34.5%		79.4%		97.0%	

Table 4 Current Magnitude comparison

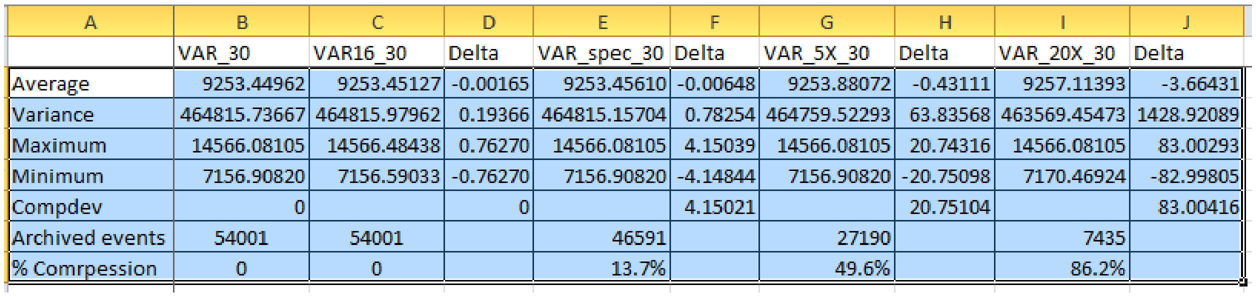
VA



A	B	C	D	E	F	G	H	I	J
	VA_30	VA16_30	Delta	VA_spec_30	Delta	VA_5X_30	Delta	VA_20X_30	Delta
Average	31763.4284	31763.4270	0.0014	31763.4992	-0.0708	31765.6848	-2.2564	31777.9052	-14.4768
Variance	882450.2707	882462.7340	0.7773	882411.5961	16.4871	881633.8320	894.8653	862104.7056	12121.7097
Maximum	38863.8594	38865.3203	1.5273	38863.8594	13.0820	38863.8594	65.4473	38863.8594	261.7676
Minimum	29360.5273	29361.8574	-1.5273	29360.5273	-13.0879	29403.2168	-65.4414	29444.3438	-261.7324
Compdev	0.0000		0.0000		13.0895		65.4476		261.7906
Archived events	54001	54001		37791		13513		2013	
% Comrpession	0	0		30.0%		75.0%		96.3%	

Table 5 Volt Amp comparison

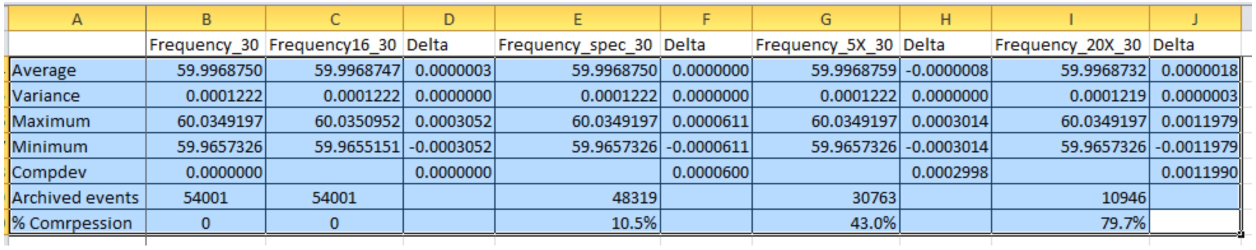
VAR



A	B	C	D	E	F	G	H	I	J
	VAR_30	VAR16_30	Delta	VAR_spec_30	Delta	VAR_5X_30	Delta	VAR_20X_30	Delta
Average	9253.44962	9253.45127	-0.00165	9253.45610	-0.00648	9253.88072	-0.43111	9257.11393	-3.66431
Variance	464815.73667	464815.97962	0.19366	464815.15704	0.78254	464759.52293	63.83568	463569.45473	1428.92089
Maximum	14566.08105	14566.48438	0.76270	14566.08105	4.15039	14566.08105	20.74316	14566.08105	83.00293
Minimum	7156.90820	7156.59033	-0.76270	7156.90820	-4.14844	7156.90820	-20.75098	7170.46924	-82.99805
Compdev	0		0		4.15021		20.75104		83.00416
Archived events	54001	54001		46591		27190		7435	
% Comrpession	0	0		13.7%		49.6%		86.2%	

Table 6 VAR comparison

Frequency



A	B	C	D	E	F	G	H	I	J
	Frequency_30	Frequency16_30	Delta	Frequency_spec_30	Delta	Frequency_5X_30	Delta	Frequency_20X_30	Delta
Average	59.9968750	59.9968747	0.0000003	59.9968750	0.0000000	59.9968759	-0.0000008	59.9968732	0.0000018
Variance	0.0001222	0.0001222	0.0000000	0.0001222	0.0000000	0.0001222	0.0000000	0.0001219	0.0000003
Maximum	60.0349197	60.0350952	0.0003052	60.0349197	0.0000611	60.0349197	0.0003014	60.0349197	0.0011979
Minimum	59.9657326	59.9655151	-0.0003052	59.9657326	-0.0000611	59.9657326	-0.0003014	59.9657326	-0.0011979
Compdev	0.0000000		0.0000000		0.0000600		0.0002998		0.0011990
Archived events	54001	54001		48319		30763		10946	
% Comrpession	0	0		10.5%		43.0%		79.7%	

Table 7 Frequency comparison

The interesting observation is that frequency does not compress as well as the other variables. This is probably because the frequency is highly variable inside the OSIsoft building. The variations are high due to spikes on the line due to the older HVAC motors in the building.

Simple Applications –Angle unwrapping

The voltage and current phasor angles are reported as discontinuous signals. This is whether the range is (-π, +π) in degrees, or (0, 2π) in radians. The physical phase angle in the power grid is normally a continuous signal, but can jump sharply during line faults or other outages. There are several methods of unwrapping phase angle: (1) computing the difference between buses first, then subtracting and assuming the two points are not islanded and (2) unwrapping the phase at each but first and then computing the angle difference. The latter method is preferred since it can be used to detect islands as well as for plotting angle surfaces. This shows the effects of three different methods of unwrapping and computing the difference between two bus voltage angles.

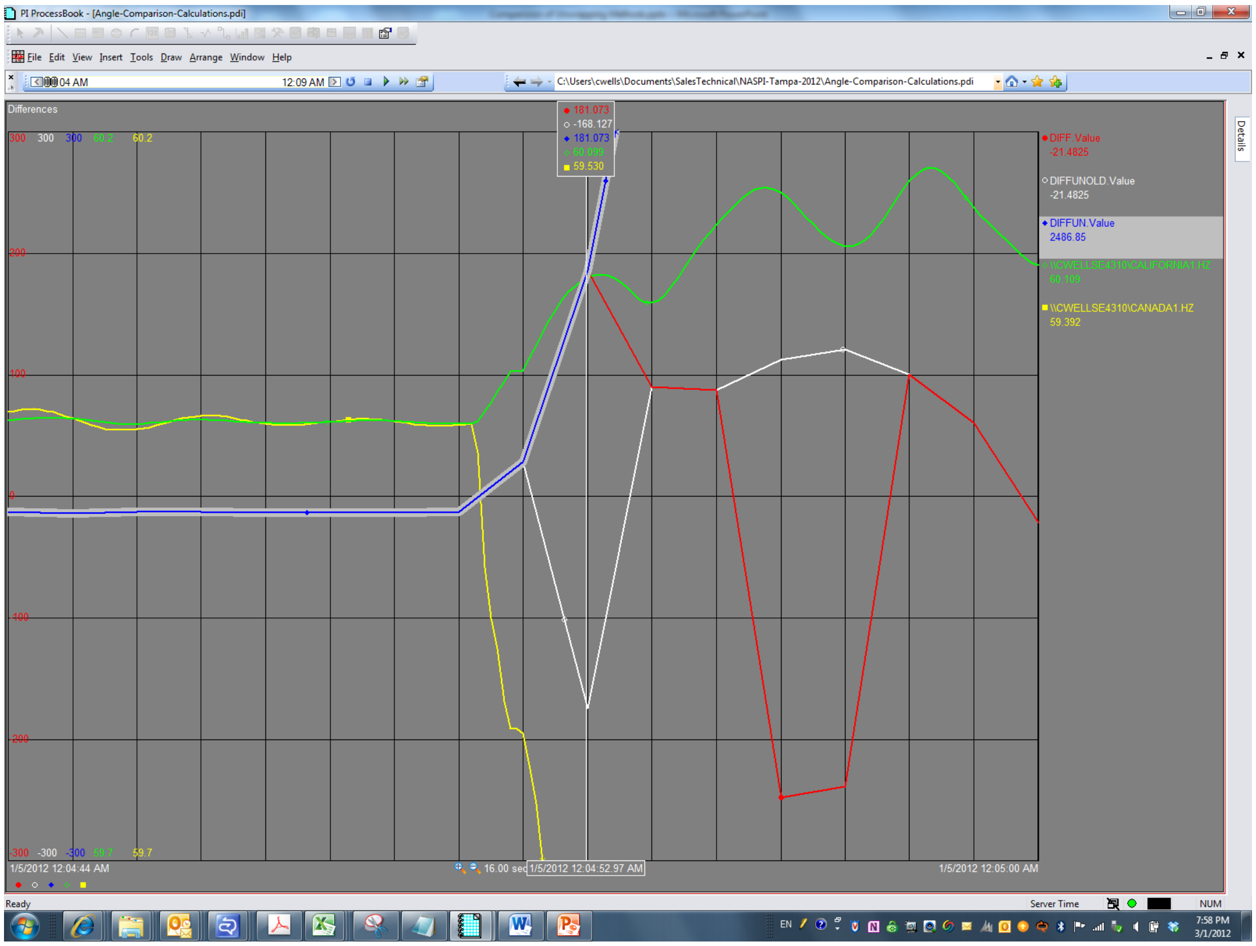


Figure 8 Comparison of three methods of angle difference calculations.

The “red” trace is the time aligned direct subtraction of two angles. The white trace is the angle difference computed assuming the maximum difference between any two angles is not greater than 360 degrees. The blue trace unwraps the individual bus angles first then performs the subtraction. This is the preferred method, since it works even when typals[1] exist in the grid.

One algorithm for the preferred approach is shown below:

Initialization:

(1) Read all angles that need to be converted to smooth variables at the exact same time instant. The time stamps must have accuracy of one micro second. Store these initial values into the initial values of the new smooth values.

[1] Ilic and Zaborszky, Dynamics and Control of Large Electric Power Networks, Wiley Interscience, 2000, page 402.

Let $As_i(j) = Am_i(j)$, where $As_i(j)$ is the smoothed value of the (i)th angle A at time (j) and $Am_i(j)$ is the measured value of the (i)th angle A at time (j). These are the initial values of the smoothed absolute angles.

Execution:

```
(2) Read all values of the absolute angles at time j+1, $Am_i(j+1)$.
    DO FOREVER
    C begin when all values at time (j+1), which is the now time (*) are in memory, this is
    similar to the way a PDC works. i.e. wait until all values at time (j+1) are in the array.
    C this code can be written so that it works on a single value rather than waiting for all
    data points to arrive in the snapshot and storing these in a memory array. Single tag
    execution will result in lower latency.
    FOR all i
    C note, it is possible to do this for each i in a separate thread.
    C compute the difference between the now value and the value at the last sample
      $\delta Am_i(j+1) = Am_i(j+1)$- $Am_i(j)$,
    C compute a trial value of unwrapped angle for all (i) values
      $As_i(j+1) = As_i(j)$+$\delta Am_i(j+1)$
    C test for a falling angle transition (increasing angle), if true correct
      IF $(\delta Am_i(j+1)) < -TestLimit)\ THEN$
            $As_i(j+1) = \quad As_i(j+1) + Range$
      ELSE
            C test for a rising angle transition (decreasing angle), if true correct
            IF $(\delta Am_i(j+1)) > TestLimit)\ THEN$
                $As_i(j+1) = \quad As_i(j+1)_i - Range$
            ENDIF
      ENDIF
    ENDFOR
    ENDDO
```

Coincidently, the unwrapped angle also provides the system time error. This is due primarily to the definition of the absolute angle from the C37.118 specification. The system reference time is at the peak of a 60 Hz cosine wave: at the top of second it has absolute angle of zero degrees and time error of zero. So by multiplying the unwrapped angle by (0.016667/360 = 0.000046), yields the time error. For example, suppose the unwrapped angle is 55 degrees; then time error is +0.002546 seconds. If the unwrapped angle is -16000; then the time error is -0.740742 seconds.

This method is an approximation since it assumes that the frequency is 60 Hz since the initialization of the unwrapping. However most ISO's make this same assumption when computing the time error. That is the number of cycles in a day is constant..

Visualization of phasor data.

There are many common ways of viewing phasor data. These include products from the major supplies of EMS systems such as ABB, Siemens, AlstomGrid, GE and OSII. Additionally other vendors such as EPG offer a visualization product called RTDMS.

The PI system also offers ProcessBook to visualize the data. A few examples of ProcessBook data visualization is shown in the following Figures.

The first example shown in Figure

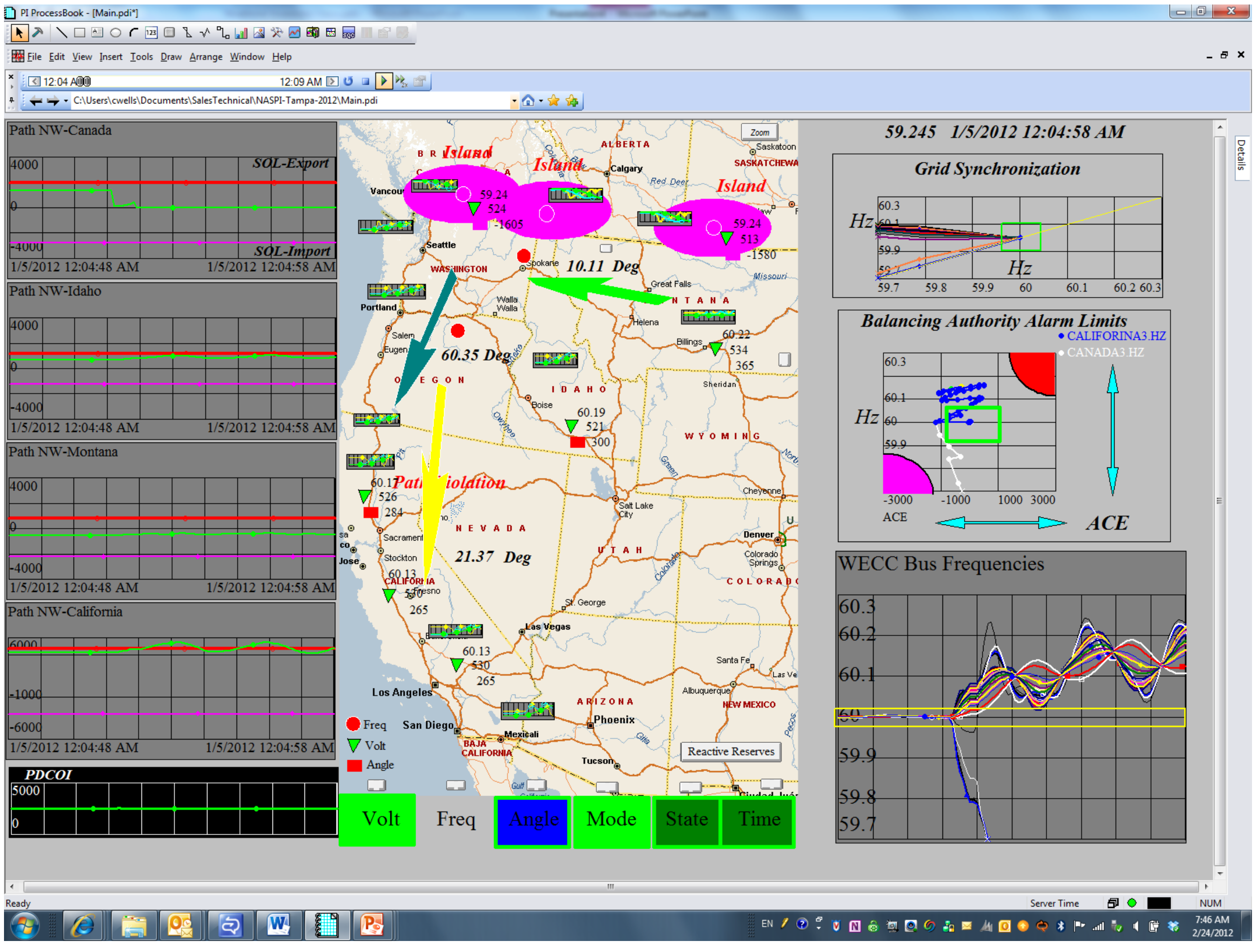


Figure 9

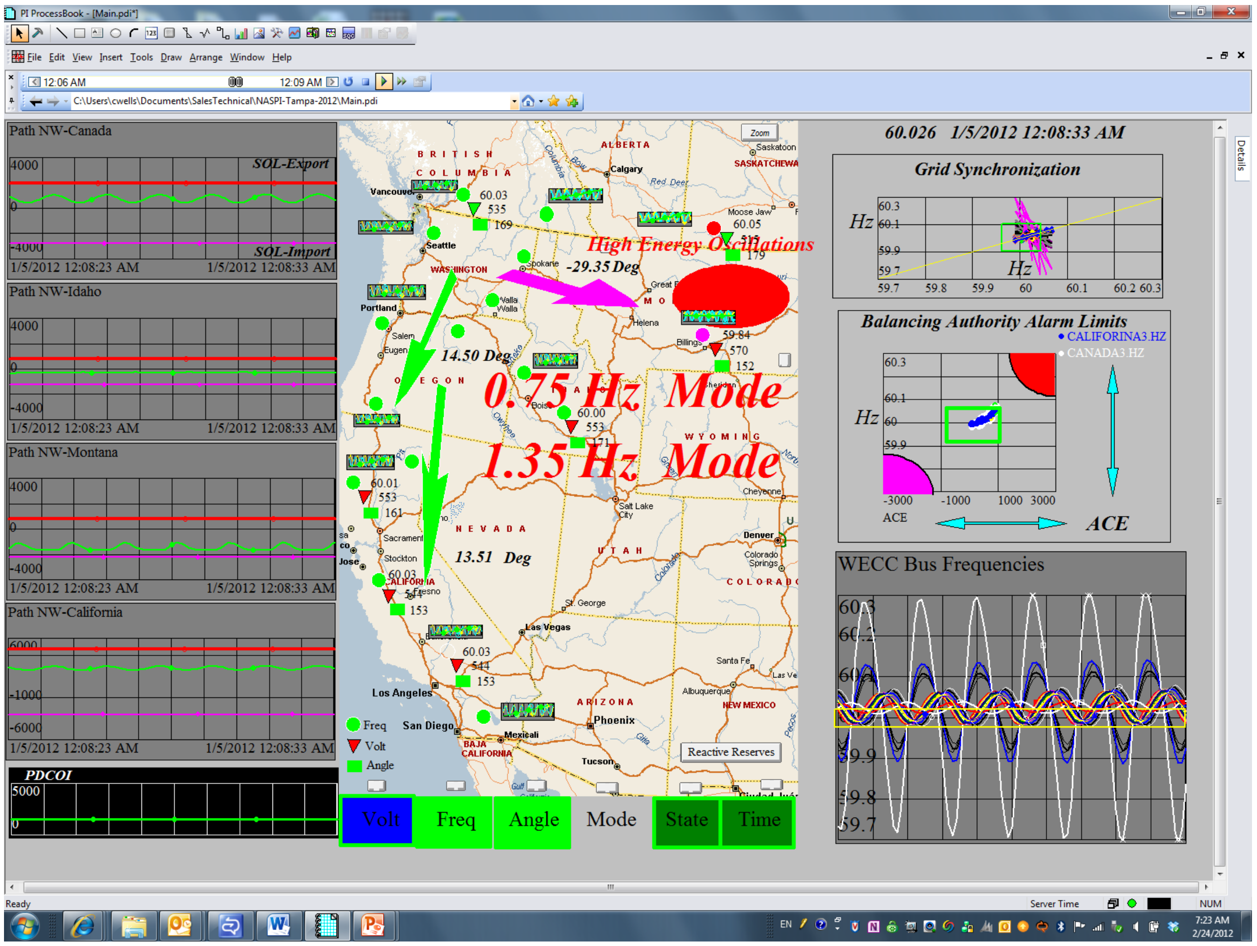


Figure 10 Unstable oscillations near central Montana

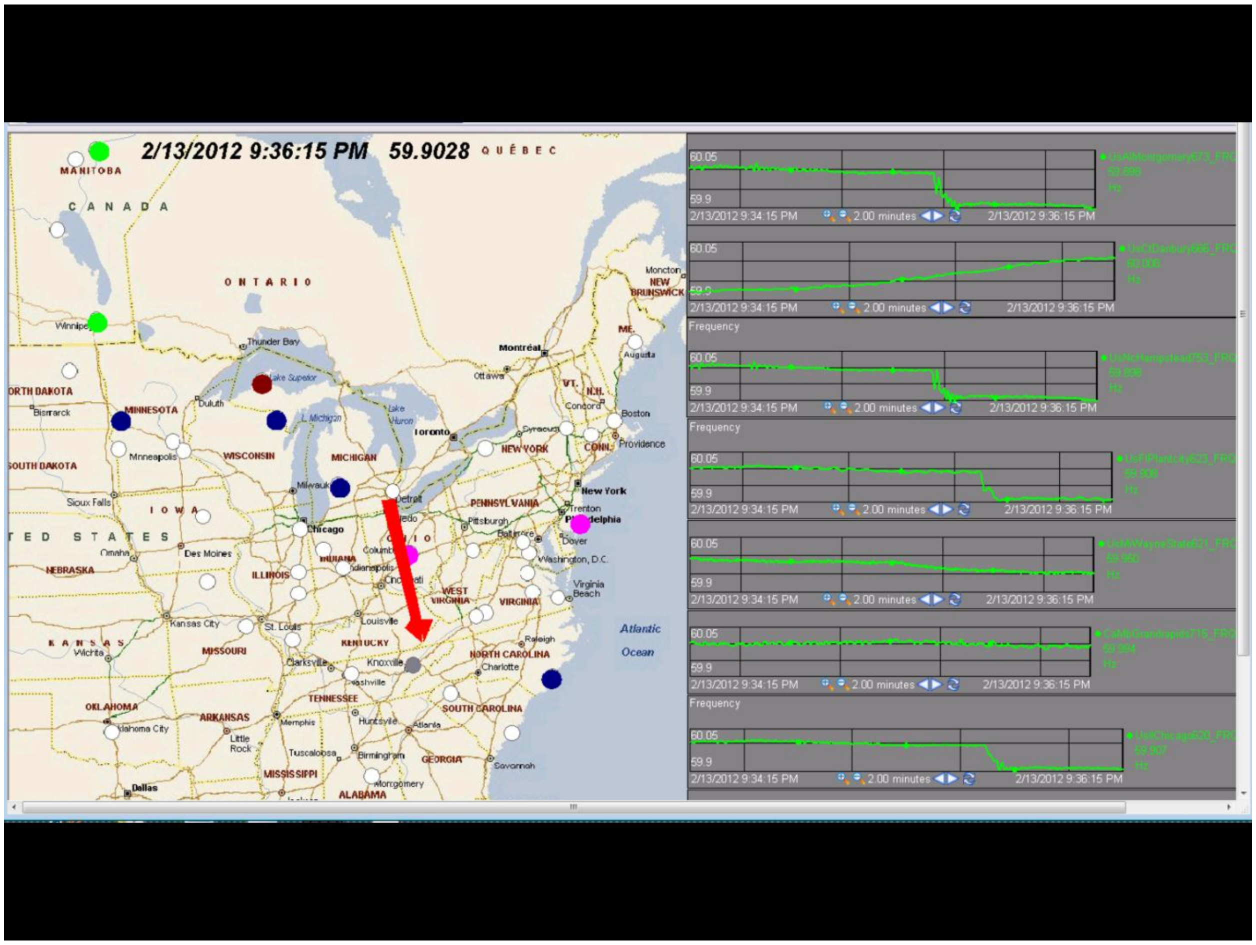
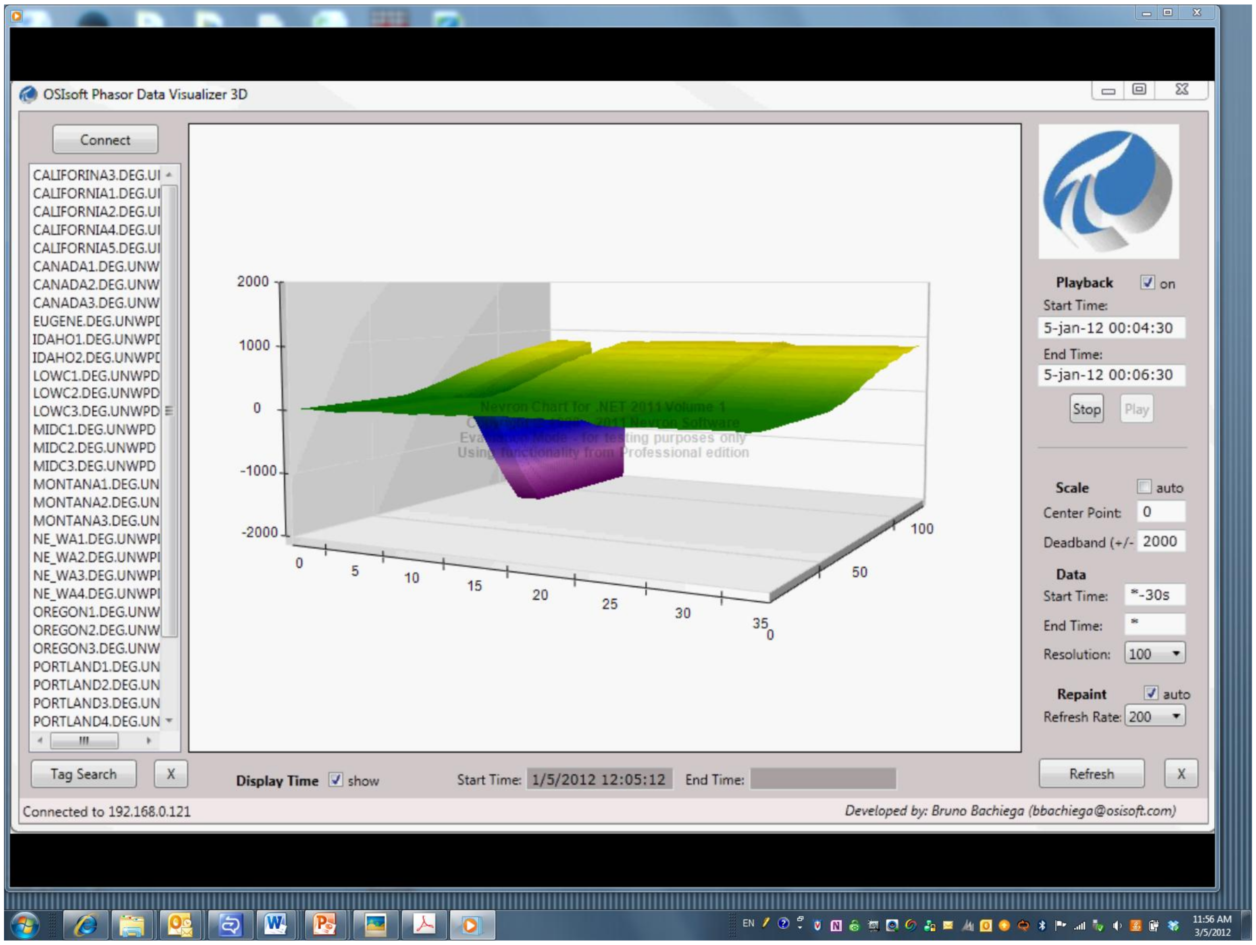


Figure 11 Frequency events in the Eastern Interconnection

An example of showing the angle surface is shown in Figure 12. This is a 3D surface of angle, time, and station number. The chart clearly shows the island formation in WECC Event 2.


OSIsoft Phasor Data Visualizer 3D
Connect
2000
1000
0
-1000
-2000
0
5
10
15
20
25
30
35
0
50
100
Nevron Chart for .NET 2011 Volume 1
2011 Nevron Software
Evaluation mode - for testing purposes only
Using functionality from Professional edition
Tag Search
X
Refresh
X
Stop
Play

CALIFORINA3.DEG.UI
CALIFORNIA1.DEG.UI
CALIFORNIA2.DEG.UI
CALIFORNIA4.DEG.UI
CALIFORNIA5.DEG.UI
CANADA1.DEG.UNW
CANADA2.DEG.UNW
CANADA3.DEG.UNW
EUGENE.DEG.UNWPD
IDAHO1.DEG.UNWPD
IDAHO2.DEG.UNWPD
LOWC1.DEG.UNWPD
LOWC2.DEG.UNWPD
LOWC3.DEG.UNWPD
MIDC1.DEG.UNWPD
MIDC2.DEG.UNWPD
MIDC3.DEG.UNWPD
MONTANA1.DEG.UN
MONTANA2.DEG.UN
MONTANA3.DEG.UN
NE_WA1.DEG.UNWPI
NE_WA2.DEG.UNWPI
NE_WA3.DEG.UNWPI
NE_WA4.DEG.UNWPI
OREGON1.DEG.UNW
OREGON2.DEG.UNW
OREGON3.DEG.UNW
PORTLAND1.DEG.UN
PORTLAND2.DEG.UN
PORTLAND3.DEG.UN
PORTLAND4.DEG.UN

Playback ✓ on
Start Time: 5-jan-12 00:04:30
End Time: 5-jan-12 00:06:30

Scale ☐ auto
Center Point: 0
Deadband (+/- 2000

Data
Start Time: *-30s
End Time: *
Resolution: 100

Repaint ✓ auto
Refresh Rate: 200

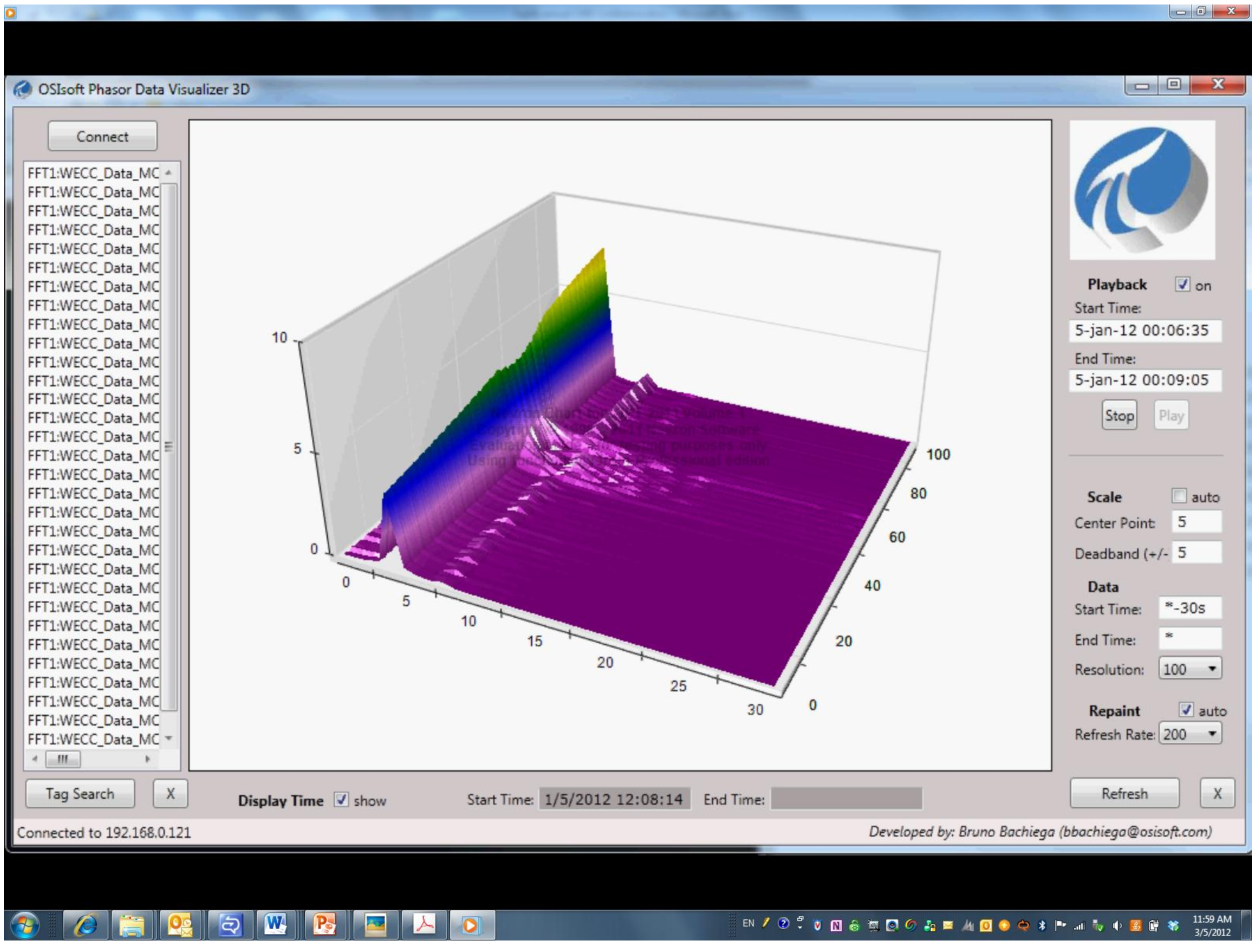
Display Time ✓ show Start Time: 1/5/2012 12:05:12 End Time:

Connected to 192.168.0.121
Developed by: Bruno Bachiega (bbachiega@osisoft.com)

11:56 AM
3/5/2012

Figure 12 Angle surface.

Another application is computing the FFT of the frequency differences between two buses. An example is shown in Figure 13.



Playback [✓ on]
Start Time: 5-jan-12 00:06:35
End Time: 5-jan-12 00:09:05

Scale [auto]
Center Point: 5
Deadband (+/- 5

Data
Start Time: *-30s
End Time: *
Resolution: 100

Repaint [✓ auto]
Refresh Rate: 200

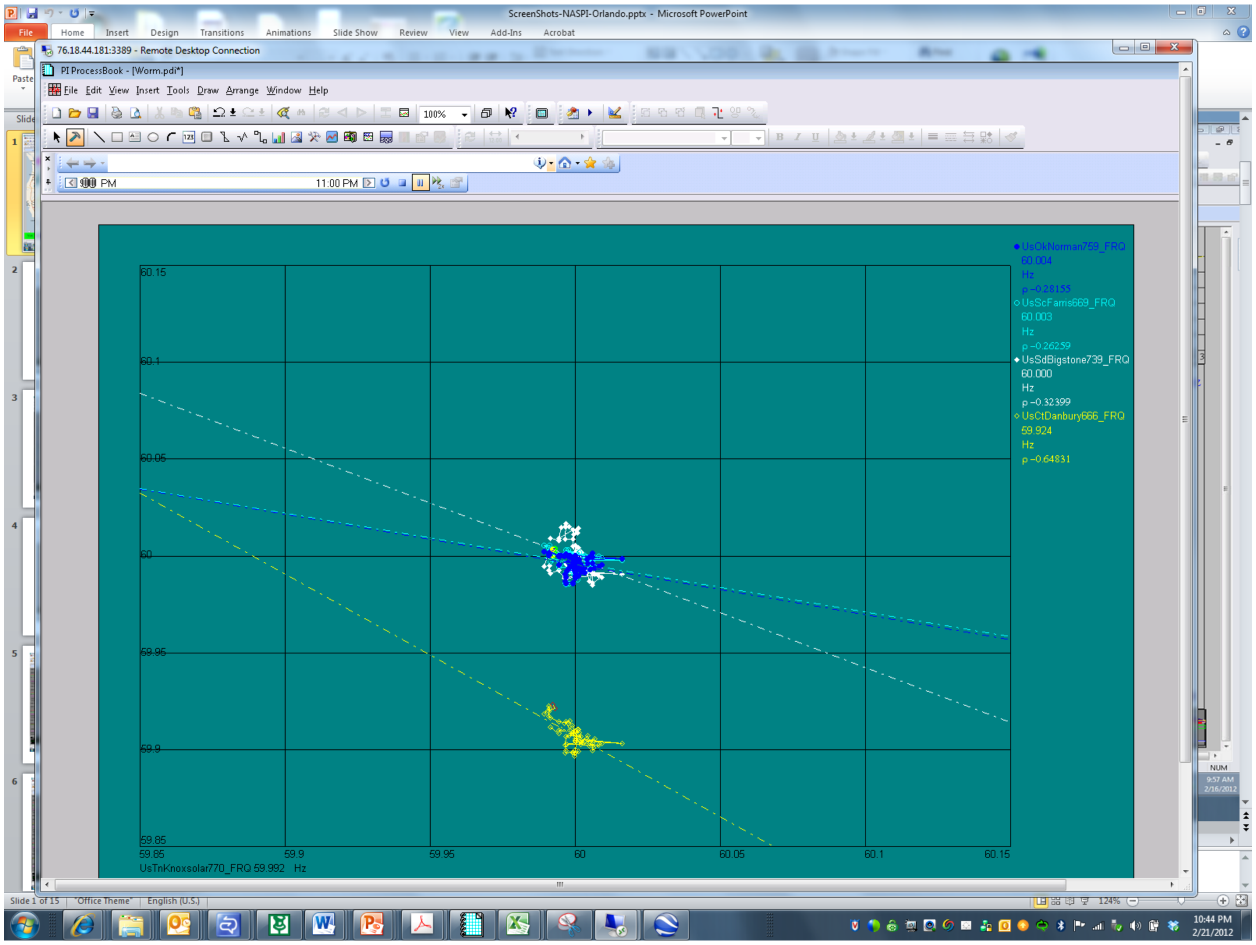
Display Time [✓ show]
Start Time: 1/5/2012 12:08:14
End Time:

Connected to 192.168.0.121
Developed by: Bruno Bachiega (bbachiega@osisoft.com)

Figure 14 FFT Waterfall chart of WECC Event 3

This chart shows the harmonic numbers on the x axis, time on the y axis and amplitude on the z axis. The FFT is computed 10 times per second on a 64 wide moving window of frequency differences between two buses in the WECC.

The following Figure shows islanding in the Eastern Interconnection. This is one form of a “state” chart since it shows the first derivative of the angle at each of six locations in the Eastern Interconnection.


ScreenShots-NASPI-Orlando.pptx - Microsoft PowerPoint
76.18.44.181:3389 - Remote Desktop Connection
PI ProcessBook - [Worm.pdi*]
File Edit View Insert Tools Draw Arrange Window Help
11:00 PM
60.15
60.1
60.05
60
59.95
59.9
59.85
59.85 59.9 59.95 60 60.05 60.1 60.15
UsTnKnoxsolar770_FRQ 59.992 Hz
● UsOkNorman759_FRQ 60.004 Hz p −0.28155
○ UsScFarris669_FRQ 60.003 Hz p −0.26259
◆ UsSdBigstone739_FRQ 60.000 Hz p −0.32399
◇ UsCtDanbury666_FRQ 59.924 Hz p −0.64831
10:44 PM 2/21/2012

The chart plots the Knoxville frequency versus four other frequencies in the Eastern Interconnection. Note that Danbury, CN is islanded from the rest of the interconnection.

Appendix A

The compression algorithm is based upon three values:

1. The most recently Archived Value
2. The current Snapshot Value
3. The incoming value to the snapshot

The determination of what to archive is based upon calculating slopes using these three values.

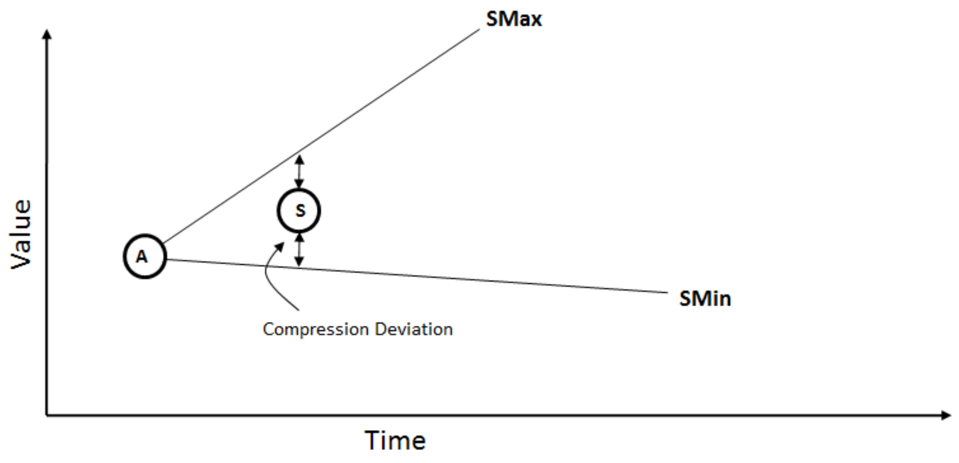
In the beginning, there is only the most recently archived value (herein referred to as the Archived Value) and the current, or Snapshot Value.

Step 1. Calculate the Slopes.

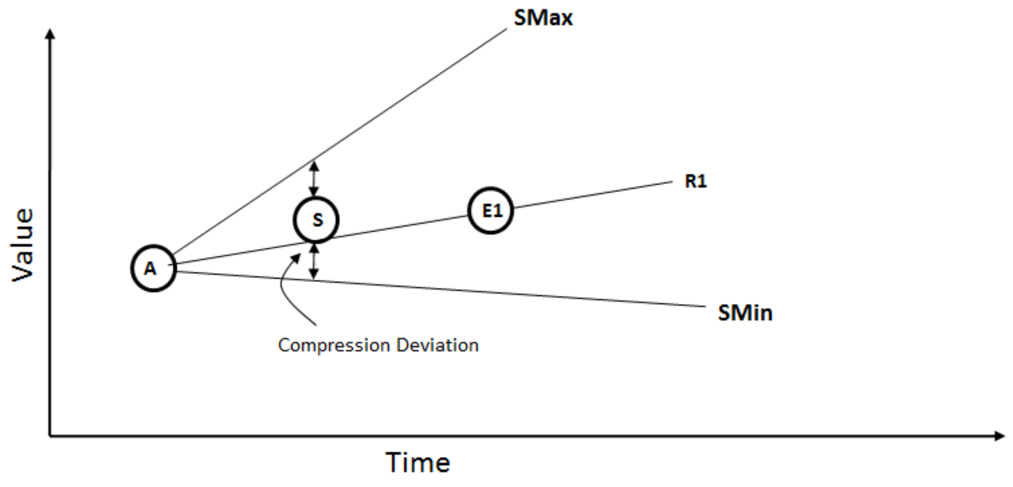
SMax = the slope of the line from the Archived Value through the Snapshot Value ***plus*** the compression deviation.

SMin = the slope of the line from the Archived Value through the Snapshot Value ***minus*** the compression deviation.

In the diagrams A represents the last Archived Value and S represents the Snapshot Value.



Step 2. Calculate and compare the Reference Slope (R1).



The Reference Slope (R) is calculated using the incoming event value (E1) and compared to SMax and SMin. The Snapshot Value is converted to an Archived Value if any of the following conditions occur:

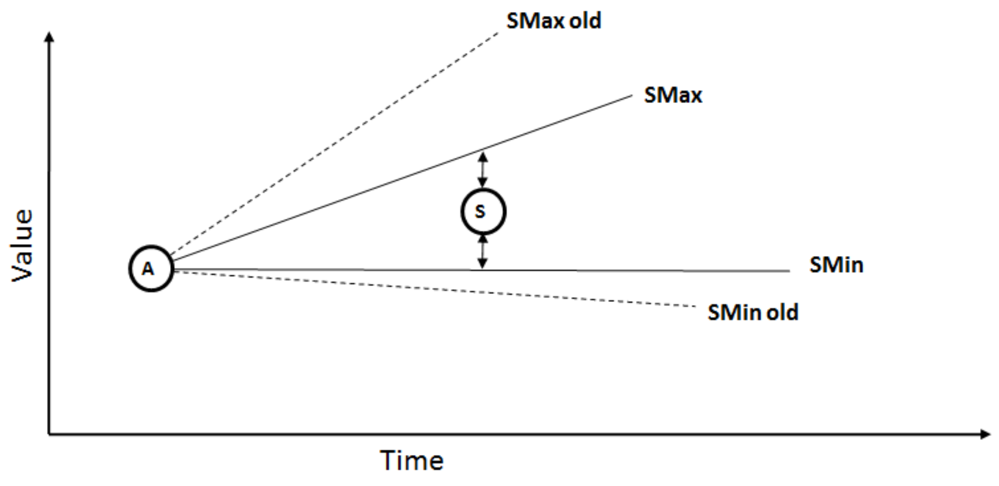
1. The Reference Slope is greater than SMax
2. The Reference Slope is less than SMin
3. The compression maximum time (not shown) has elapsed

If none of these conditions occur the following are considered true:

1. the Reference Slope is found “inside” the interior angle formed by SMax and SMin
2. the incoming value is retained as the new Snapshot Value
3. the original Snapshot Value is discarded.

In our example here R1 was inside the area formed by SMax and SMin so the new value became the Snapshot Value and the old snapshot value is discarded.

Step 3. Recalculate Min/Max Slopes. New SMax and SMin slopes are recalculated using the new snapshot value and the compression deviation values.

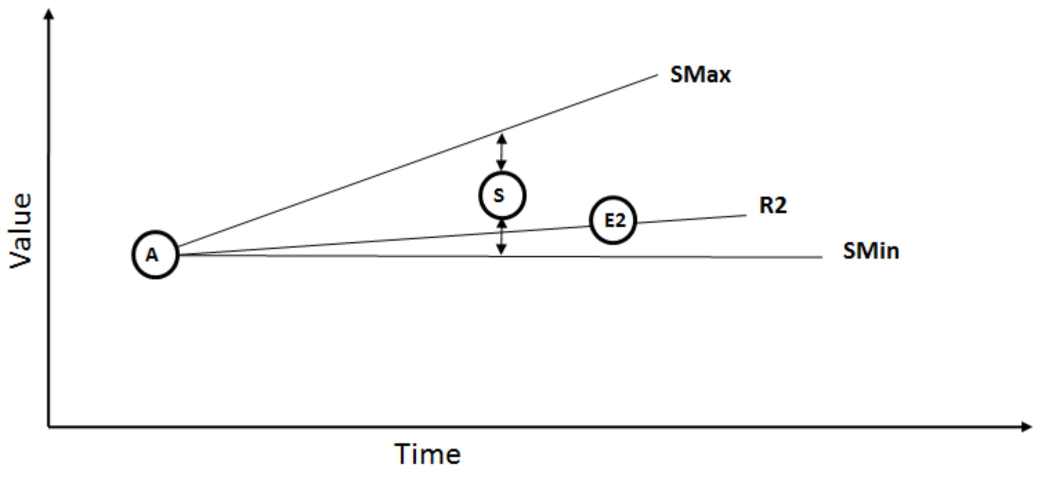


There is one additional rule applied. ***The slopes calculated as the new SMax and SMin must never expand from their previous calculated values***. Therefore, if a new slope for SMax or SMin are “outside” the previous then those new slopes are discarded in favor of the old. The inside angle formed by the slopes must always narrow, or we could get into a situation where nothing would ever deviate.

In our example above the new slopes are “narrower” than the old slopes so we retain them.

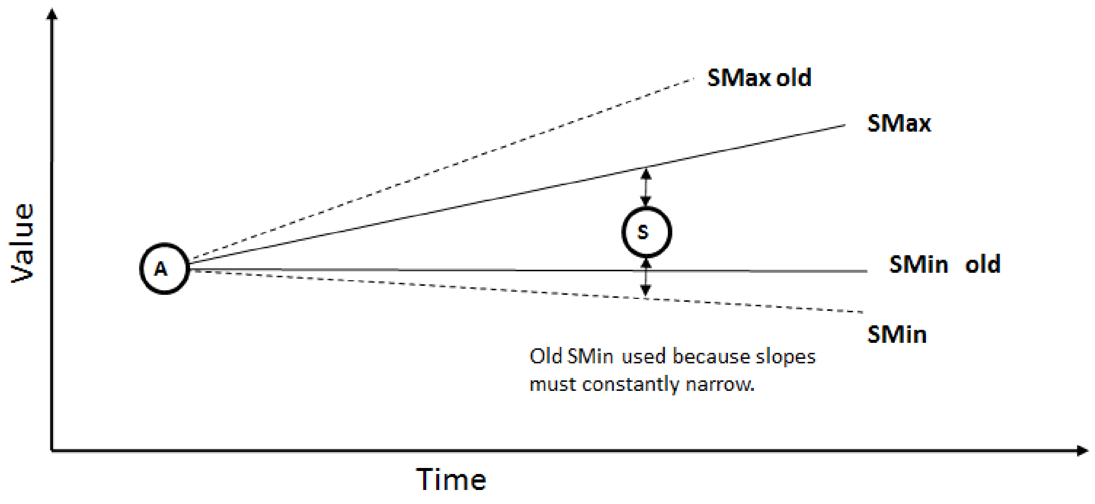
Step 4. Repeat Comparison and Slope Calculation

Again the Incoming Event Value (E2) is used to calculate the Reference Slope (R2), which is compared to SMax and SMin.

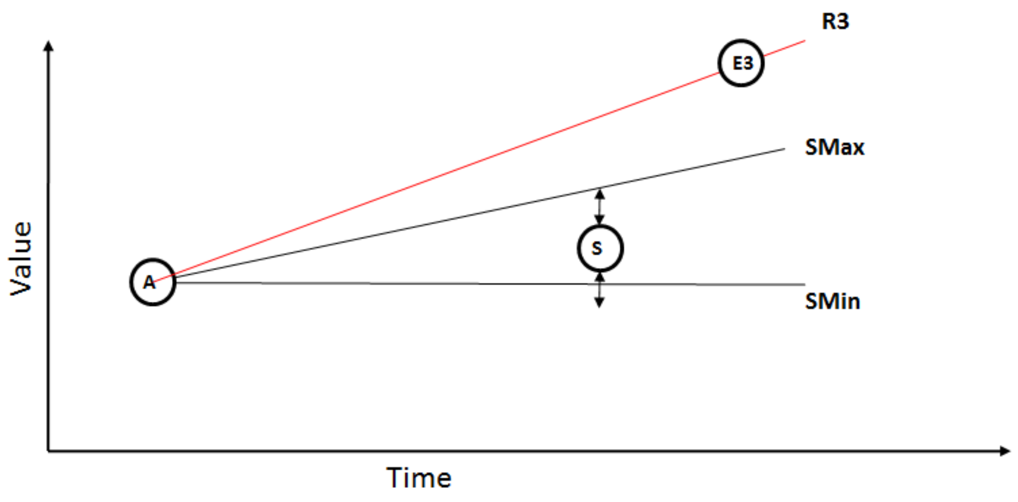


It does not fall outside the slopes so we retain the event as the new snapshot and repeat.

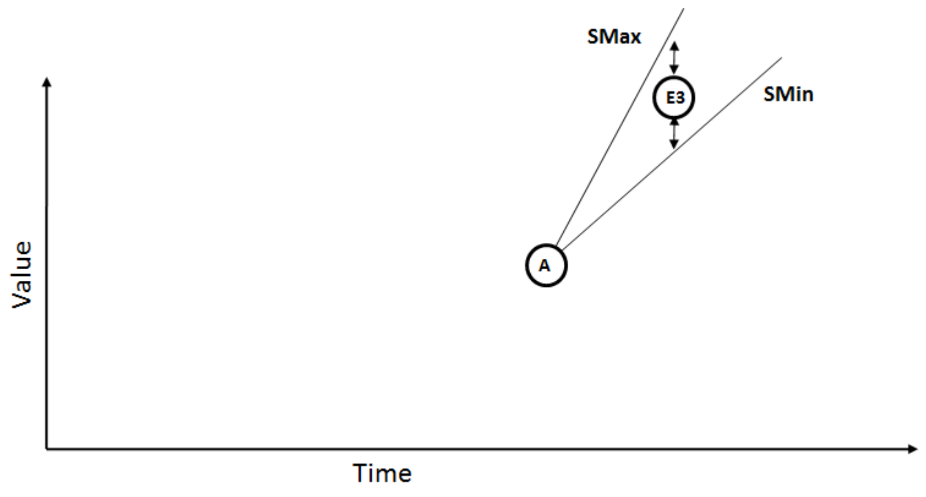
In the diagram below we see that after calculating the new slopes the SMin is less than the previous SMin. In this case we will use the new SMax and the previous SMin for the next comparison.



Step 4. New Archived Value. We will eventually calculate a Reference Slope that will be outside SMax and SMin. In the diagram below the third event value (E3) has a Reference Slope (R3) that is greater than SMax.



When this occurs the old Snapshot value becomes the Archived Value and the incoming value becomes the Snapshot Value. The process is reset and the SMin/SMax slopes are recalculated accordingly.



Conclusion

The archiving process is designed to present values to the Archive that exceed the Compression Deviation. If we were to draw an imaginary parallelogram using the last archived value and the latest snapshot value that arrived subsequent to that but before the next deviation, we would find the values received to be "inside" that parallelogram as shown in the main body of the paper.