# Phase Angle Calculations:

## Considerations & Use Cases

KEN MARTIN, EPG
EATT PHASE ANGLE PAPER TEAM
NASPI WORK GROUP MEETING
OCTOBER 2016

# Acknowledgment


- Jeff Anderson, *BPA*
- Dan Brancaccio, *Bridge*
- Jeff Dagle, *PNNL*
- Harold Kirkham, *PNNL*
- Jim Kleitsch, *ATC*
- Dmitry Kosterev, *BPA*
- Ken Martin, *EPG*
- Jim McNierney, *NYISO*
- Vincent Minden, *BPA*
- Ryan Nice, *PJM*
- Ryan Quint, *NERC*
- Alison Silverstein, *NASPI*
- Kyle Thomas, *DVP*
- Mani Venkatasubramanian, *WSU*
- Chuck Wells, *OSIsoft*

# Phase Angle Formulation

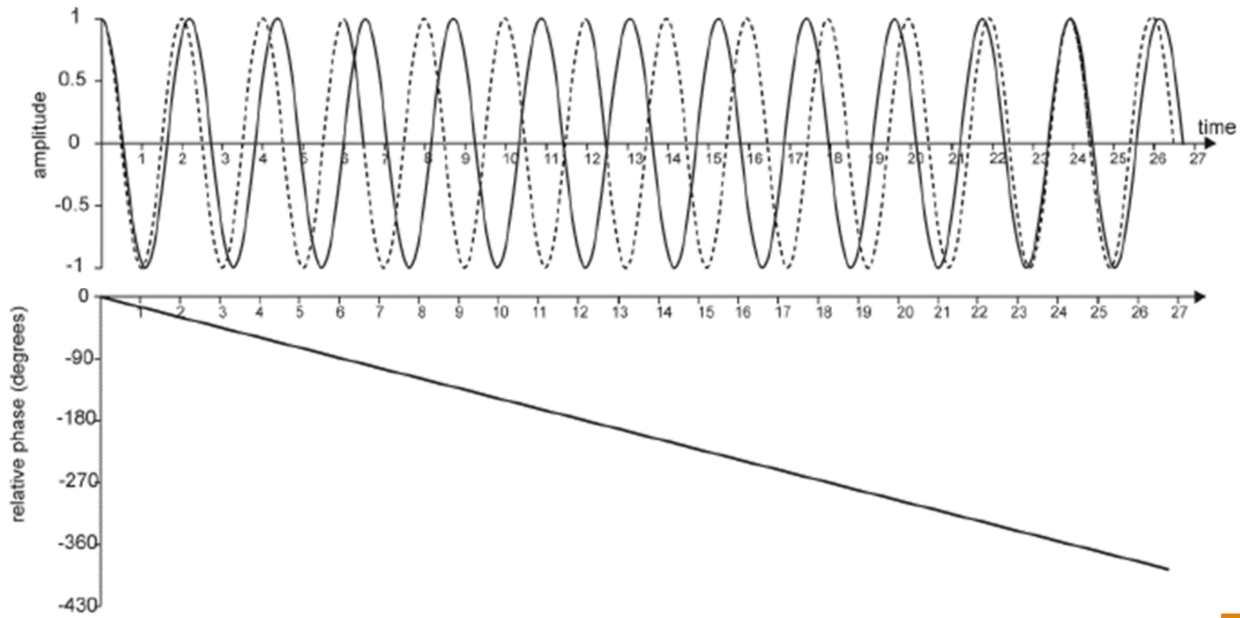
- PMU reports measured values time-tagged to UTC
  - Amplitude
  - Phase Angle
  - Frequency
  - Rate of change of frequency (ROCOF)
- Signal can be modeled as *x(t)* and the Synchrophasor is calculated relative to the nominal system frequency synchronized to UTC:

$$x(t) = X_m(t)\cos(2\pi f_0\, t + \varphi(t))$$

- The synchrophasor is: $\boldsymbol{X}(t) = (X_m(t)/\sqrt{2})e^{j(\varphi(t))}$
  - The synchrophasor includes all the dynamic changes in amplitude and phase angle
  - Off-nominal frequency appears as a change in phase angle

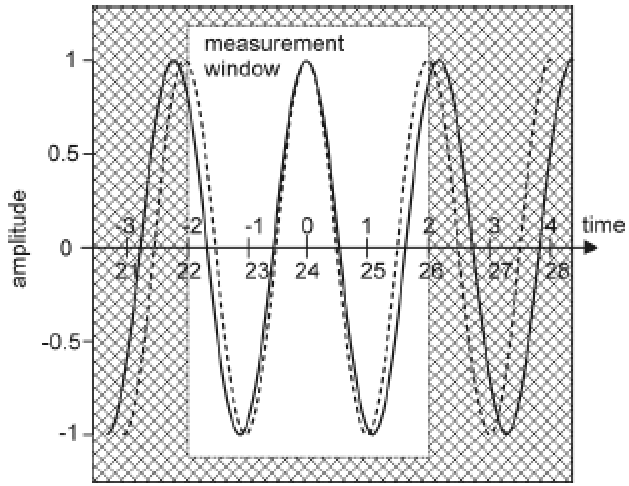
# Frequency & Angle Example

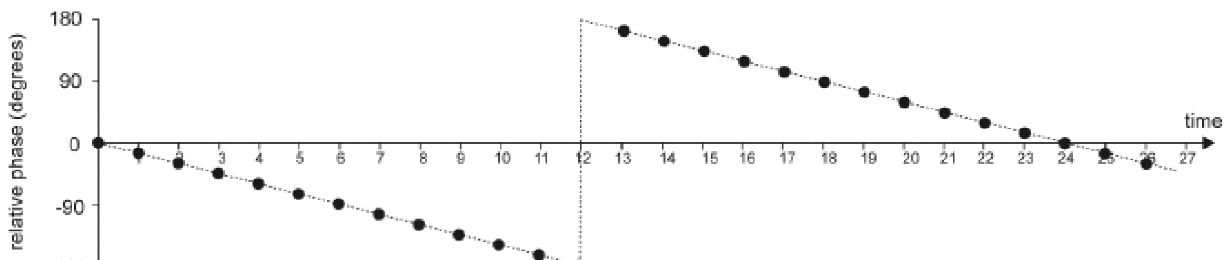
- Off-nominal frequency $f = f_0 + \Delta f$:
- Signal formula is: $x(t) = X_m \cos(2\pi(f_0 + \Delta f)\, t + \varphi) = X_m \cos(2\pi f_0\, t + \underbrace{2\pi\Delta f\, t + \varphi}_{\phi(t)})$
- Then the synchrophasor is: : $\boldsymbol{X}(t) = X_m/\sqrt{2}\, e^{j(2\pi\Delta f\, t + \varphi)}$
- Constant *f* off-nominal results in angle increasing/decreasing linearly



# PMU Window Measurement

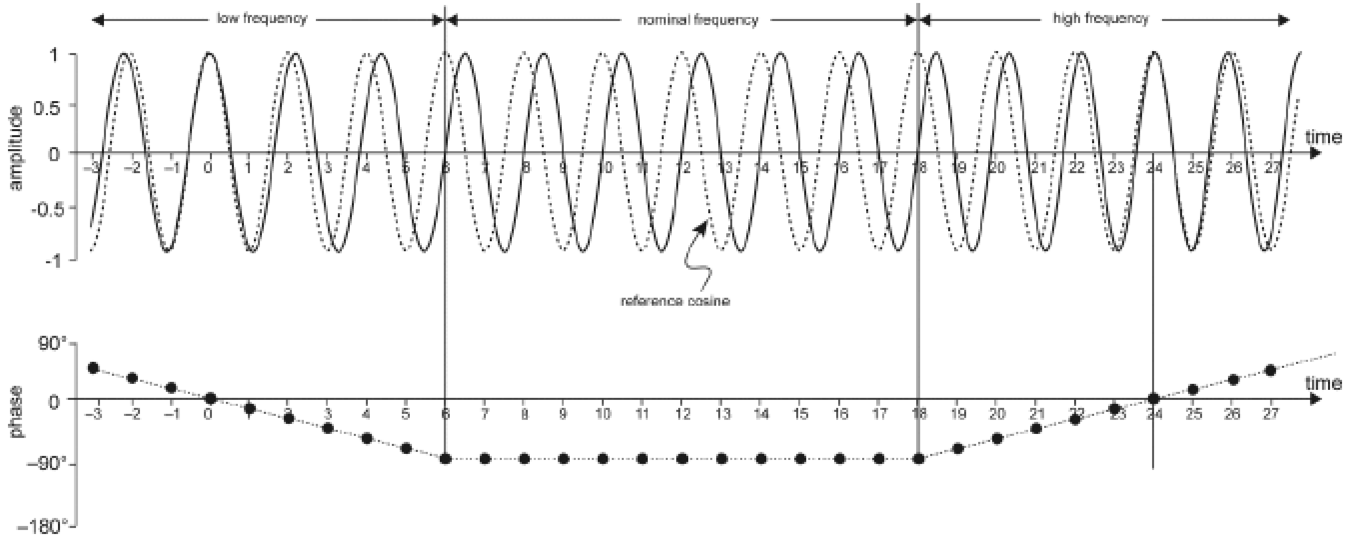
- A PMU examines the signal over short time (measurement window)
  - Generally a few cycles
- It estimates the phase angle at each point and resolves the answer to a principal value θ where $-\pi < \theta \leq \pi$.
  - The PMU only estimates phasor angle, it does not track time error
  - In a time plot, PMU may appear to “wrap” but that is not something a PMU does… It is an artifact of the way we resolve angle to the principal value



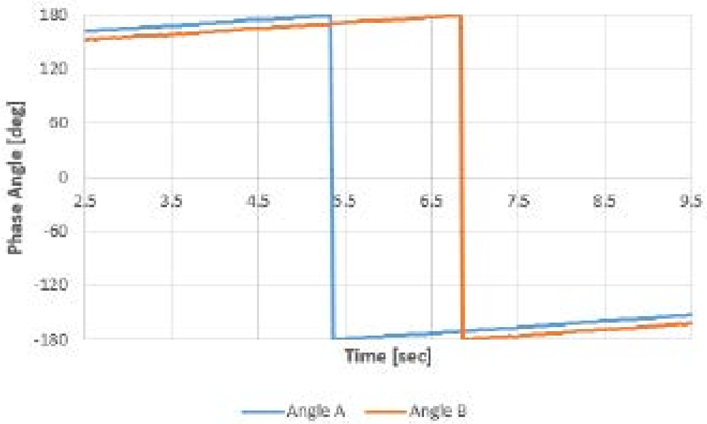
# Accumulated Phase

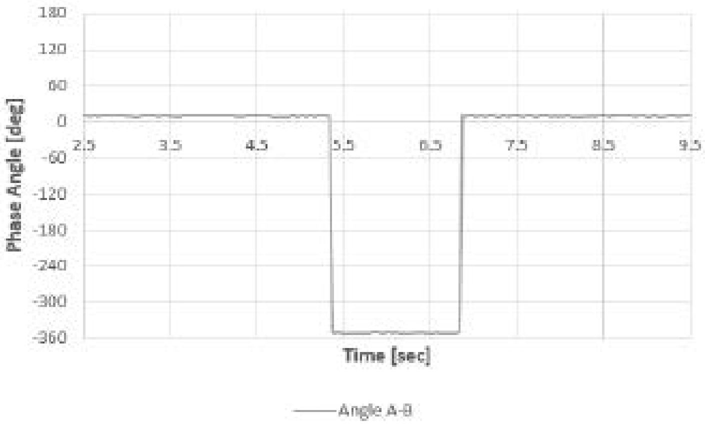
- As frequency varies off-nominal, phase angle will vary
  - System frequency > nominal, the phase angle increases
  - System frequency < nominal, the phase angle decreases
  - Phase angle may decrease for period of time and then increase for some time
- Accumulated phase can be tracked based on some starting point



# The Issues

- To calculate phase angle between two measurements, we have to be sure they do not span the 180° rollover
- As they rotate with off-nominal frequency, they do not roll-over at the same time and create a spike in the measurement
- We may want to record the accumulated system time error through the accumulated phase angle

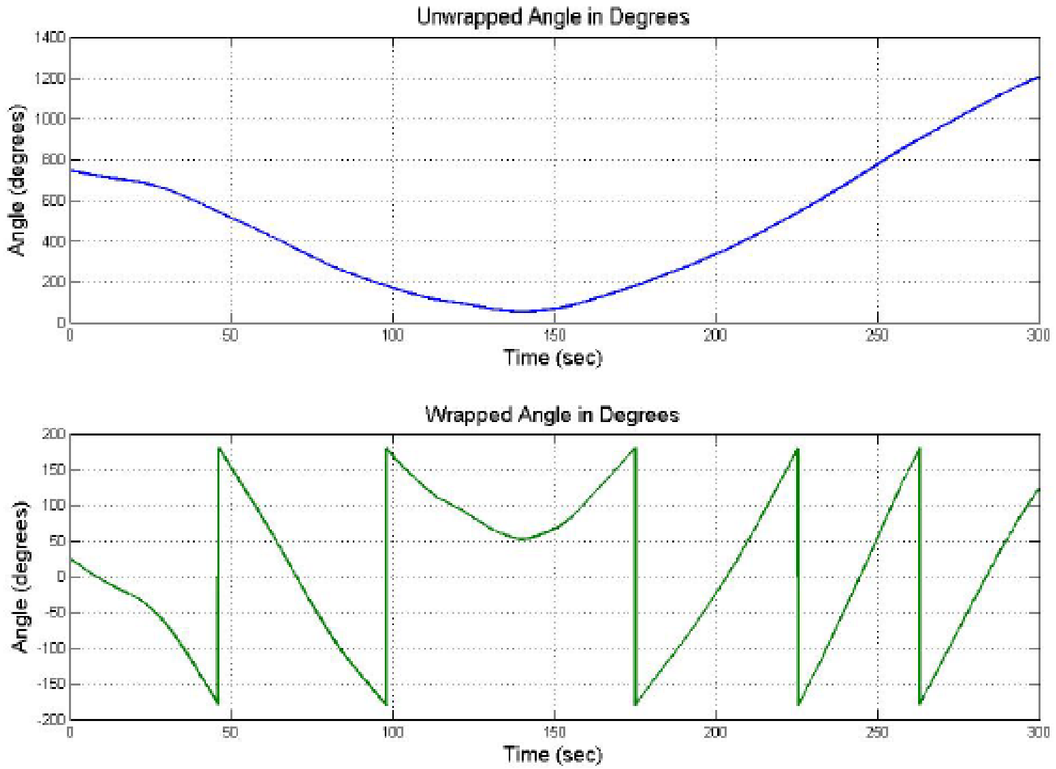


# Solution approaches

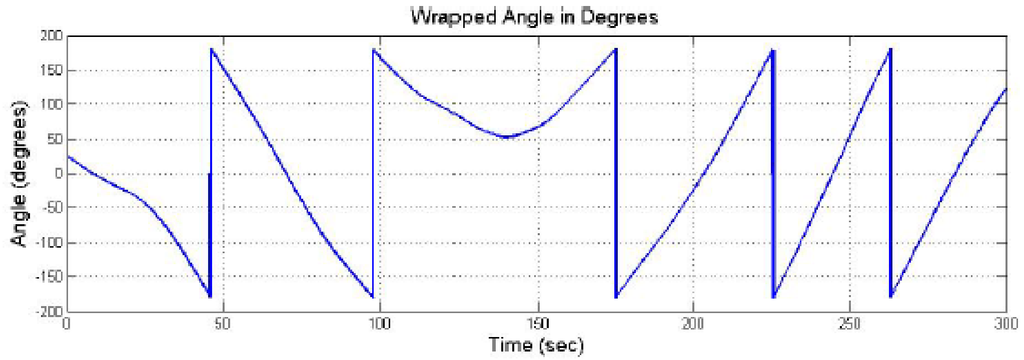
- Each time angle differences are calculated, resolve them into their principal value (subtract then unwrap)
- Monitor each angle from some starting point and add the full circle (360°) to an accumulated phase angle value each time it rolls over (unwrap the angle). Calculate all angle differences from these unwrapped angles
  - Starting points have to be unwrapped initially
- Both approaches assume there will be no real angle differences that are greater than 180°. In fact, given any two angles, it is impossible to determine angle differences >180° without additional information
  - Angles at intermediate points
  - Flow calculations with equipment and line parameters (state estimation)

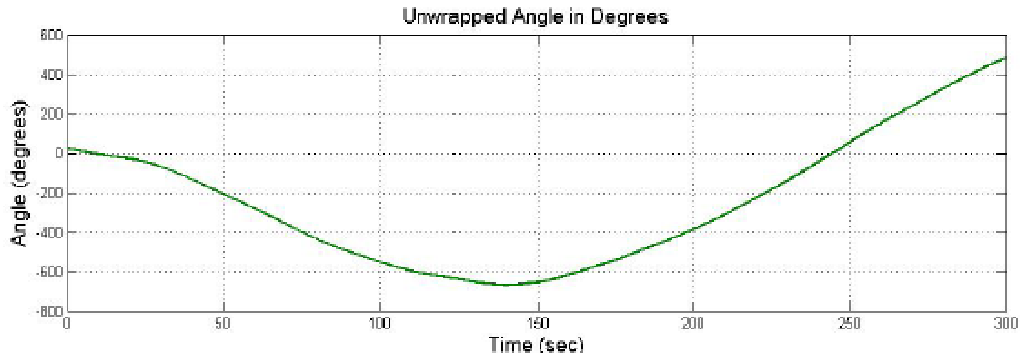
# Angle Wrapping Definition




$$\Omega(\varphi) = \begin{cases} \varphi - 2n\pi, \text{ where } n \in I \text{ is an integer so that } -\pi < \varphi - 2n\pi < \pi \\ \pi, \text{ where } n \in I \text{ is an integer so that } \varphi - 2n\pi = \pi \end{cases}$$

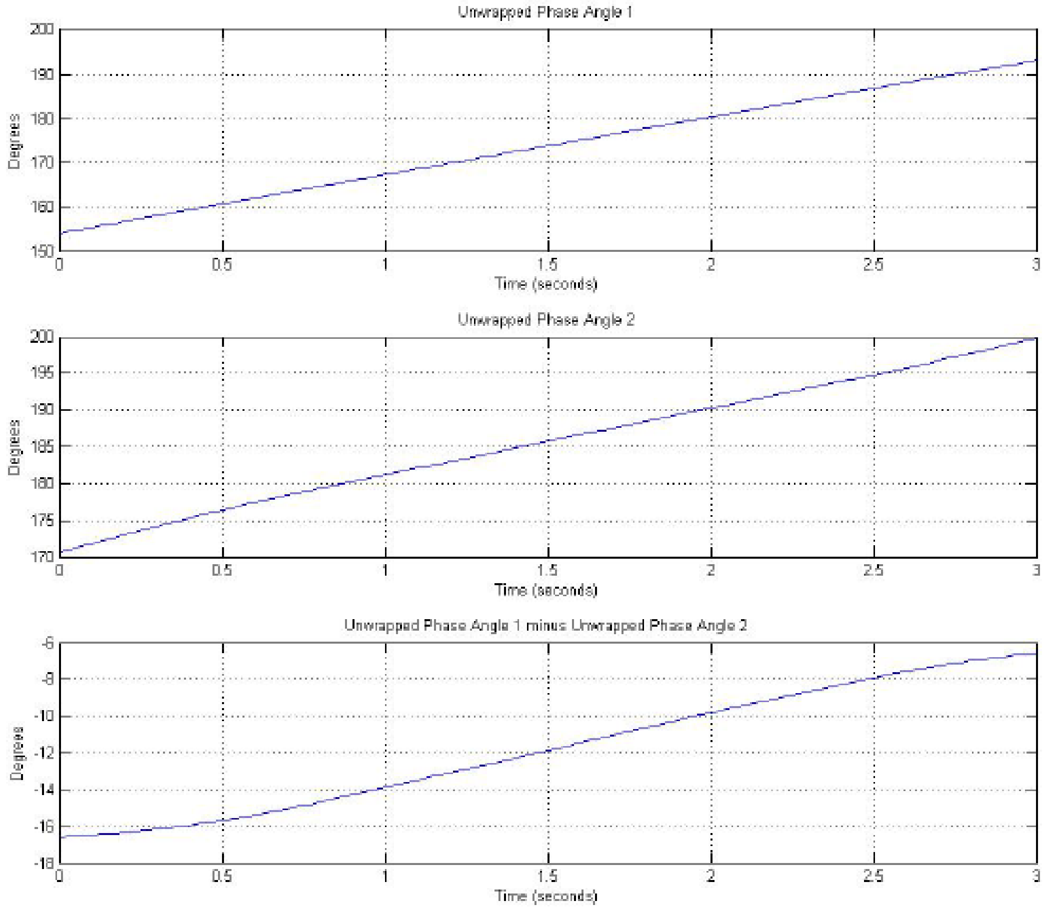
# Angle Unwrapping Definition



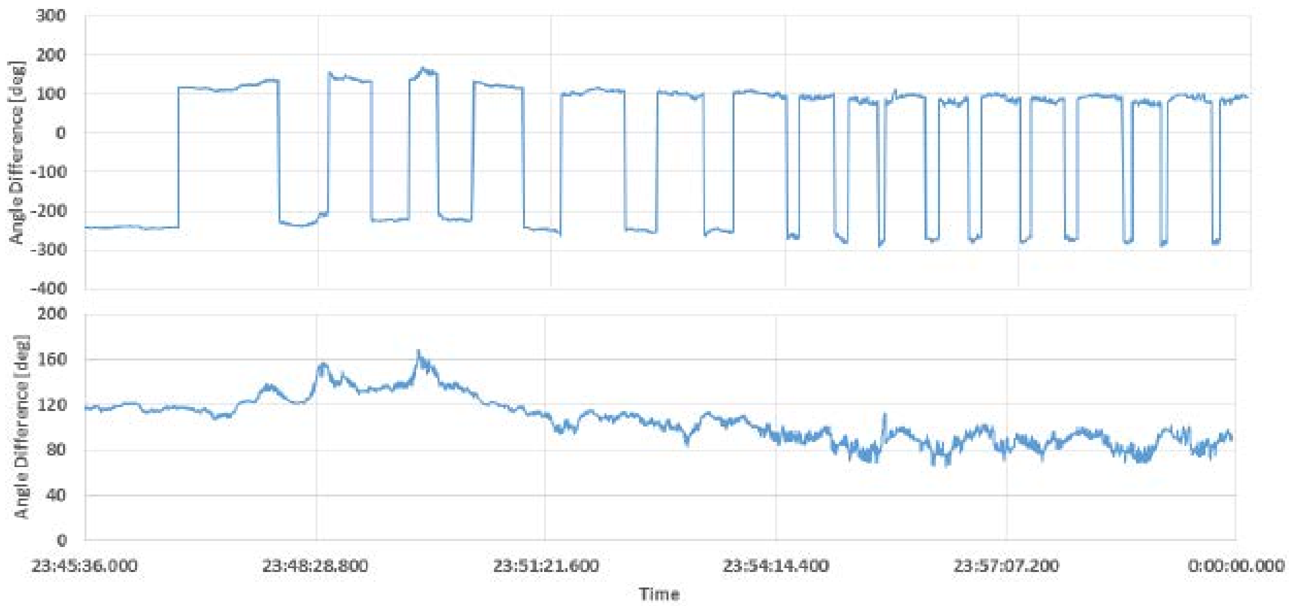

$$\begin{cases} \Psi\big(\phi(t_0)\big) = \phi(t_0), \\ \Psi\big(\phi(t_k)\big) = \phi(t_k) + \Psi\big(\phi(t_{k-1})\big) - \phi(t_{k-1}) + 2n\pi, \text{ where } n \text{ is an integer} \\ \qquad\qquad \text{that minimizes } |\phi(t_k) - \phi(t_{k-1}) + 2n\pi|, \ \forall\, k \neq 0. \end{cases}$$

# Unwrap Then Subtract



# Subtract Then Wrap

```
double UnwrapAngleDelta(double AngleDelta)
{
    double rval = AngleDelta;
    if (Math.Abs(AngleDelta) > 180)
    {
        if (Math.Sign(AngleDelta) < 0)
is AngleDelta negative?
        { rval = AngleDelta + 360; }
        else // AngleDelta is positive
        { rval = AngleDelta - 360; }
    }
    return (rval);
}
```



# Vector Math

This approach lets the computer math formulation resolve the angle. This works because vector (and complex number) math works in rectangular coordinates which do not have a roll over and resolves the answer to the principal value.

$$\alpha_{Diff} = \angle\left(\bar{V}_1 \cdot \bar{V}_1^{\,*}\right)$$

```
V1 = 1+0i;
V2 = 1+1i;
dDiff = angle(V1*conj(V2))*180/pi  % in degrees
```

# Phase Angle Difference Calculation Considerations

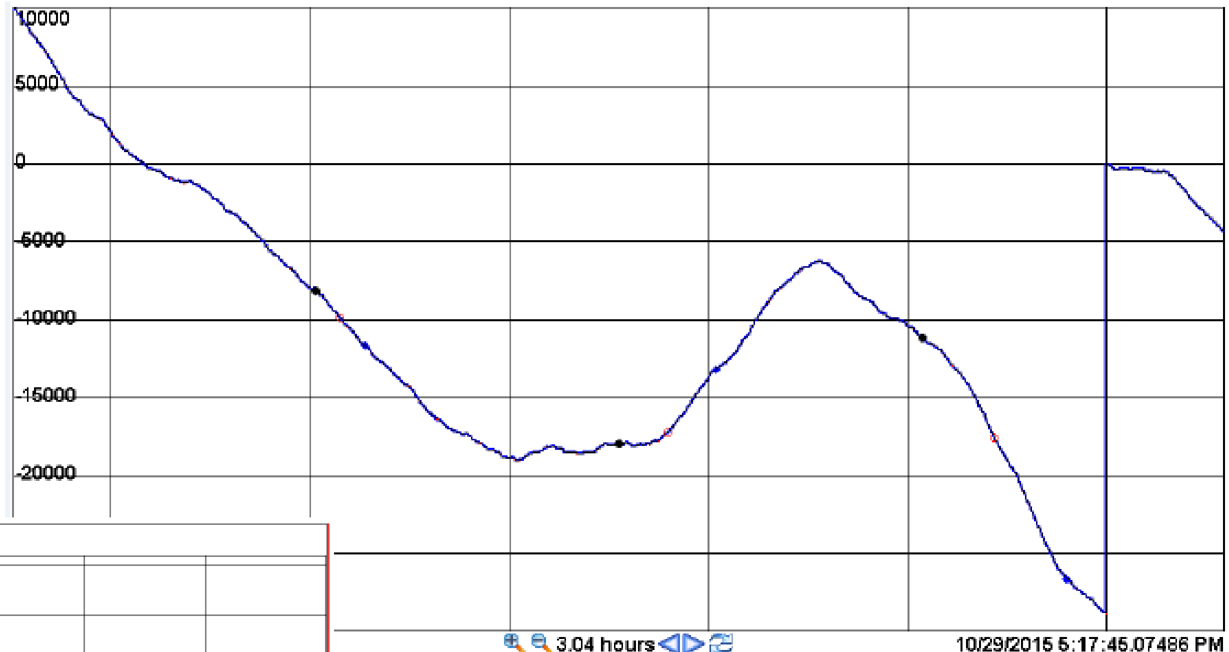
| | **Simple Arithmetic Subtraction** | **Unwrap Then Subtract** | **Subtract Then Wrap** | **Vector Mathematics** |
|---|---|---|---|---|
| EMS Calculation Capabilities | Achievable | EMS computation limitations | Achievable | Not possible |
| Calculation Accuracy | Poor, when raw angles wrap | Accurate calculation. Same level of accuracy. | | |
| Calculation Speed | Fast | | | Slow |
| CPU Usage | Minimal | | | |
| Data Storage | Minimal | High | Moderate | Minimal |
| Missing Data Issues | Minimal issues; only calculations with missing data voided | Missing data can result in wrap around points getting missed | Minimal issues; only calculations with missing data voided | Minimal issues; only calculations with missing data voided |
| Angle Resetting | None | User can specify rate of resetting[4] | None | None |
| Application Requirements | Not acceptable | Acceptable | | |
| Absolute Time Error Correction | Not achievable | Required | Not achievable | Not achievable |

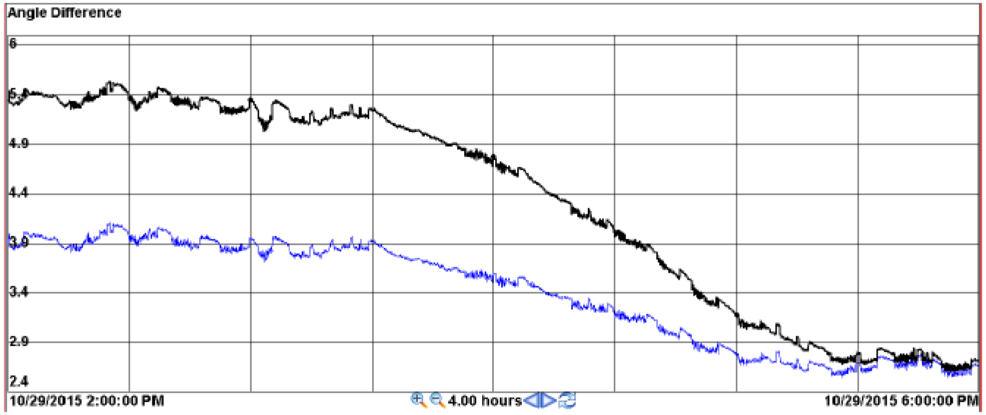
# Accumulated Phase Angle Difference

- Phase angle reported by a PMU is defined as the principal value (PV) of the signal relative to the reference waveform synchronized to UTC.
- PMU is not designed to keep track of evolution of relative phase of two signals since some arbitrary time zero.
- A PMU could be built to track accumulated phase
  - The user would have to establish a start time
  - Accumulated phase can become arbitrarily large, so a reset mechanism is required
  - Phasor value reporting would need modification to report very large values
- Accumulated phase can be calculated or obtained from the PMU signals by "unwrapping" the measurement over a period of time.
- This is effectively an alternative to calculating system time error used in SCADA systems

# Phase Angle Distance

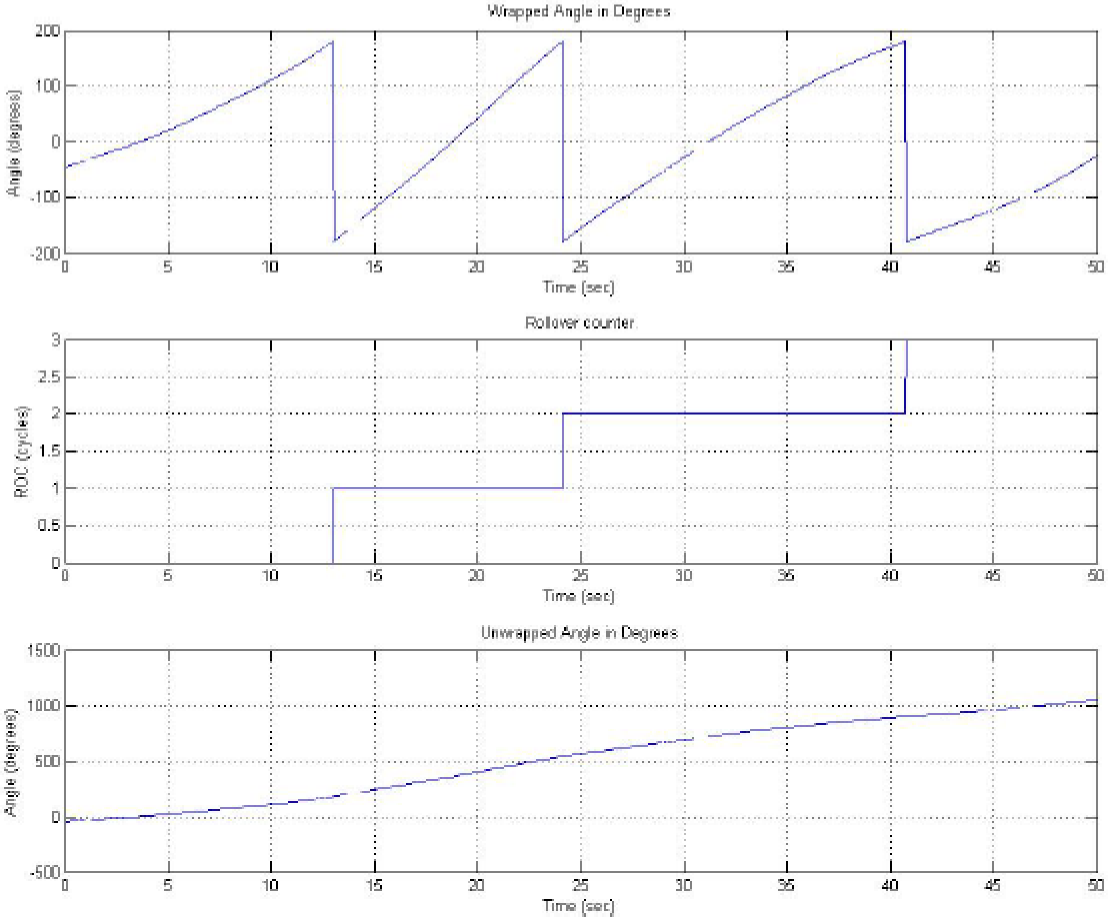
Shows how phase angle can increase to large values over time



# Rollover Counter (ROC)

For keeping track of the number of roll-overs that have occurred



# Concluding Remarks

- Applications must account for principal value of phase angle reported by PMU resolving to interval (+180,-180] in any calculation
- Applications should document and clearly define angle wrapping and unwrapping they use for industry reference
- Commonly used angle difference calculation methods explained in detail.
  - Vector math is "gold standard" – computationally complex but useful for offline applications
  - "Subtract then Wrap" and "Unwrap then Subtract" both accurate and their pros and cons
  - Simple arithmetic difference should be avoided
- Considerations include speed, accuracy, CPU usage, data storage, missing data, etc.
- Phase Angle Distance and Rollover Counter (ROC) introduced