PDV Uncertainty Estimation
& Method Comparisons

Eric Machorro, NSTec
PDV Workshop at LLNL
November 2-3, 2011

This work was done by National Security Technologies, LLC, under Contract No. DE-AC52-06NA25946 with the U.S. Department of Energy.
Abstract

Several methods are presented for estimating the rapidly changing instantaneous frequency of a time varying signal that is contaminated by measurement noise. Useful \textit{a posteriori} error estimates for several methods are verified numerically through Monte Carlo simulation. However, given the sampling rates of modern digitizers, sub-nanosecond variations in velocity are shown to be reliably measurable in most (but not all) cases. Results support the hypothesis that in many PDV regimes of interest, sub-nanosecond resolution can be achieved.

1. Compare a few methods & serve as a plug for a methods comparison report
2. Show some error bar estimates that quantify some of the uncertainty in PDV calculations.
Generating the Standard Spectrogram

- For each selected time interval of length $N$
  - Multiply measured signal by cosine window.
  - Calculate magnitudes of Fourier coefficients of result.

- Perform for intervals centered at $N/2$, $N/2+1$, $N/2+2$, …

- Result is a series of vectors of Fourier coefficients
  - Each has length $N/2$ (toss out the negative frequencies)
  - Time spacing of vectors is $1\Delta t$

- Plot the result (spectrogram)

CPU time is cheap enough so that there isn’t a need to pick the “optimal” slide length.
Find Peaks by *correctly* interpolating (IpFFT)

- For each vertical column, find pixel with maximum intensity.
- Interpolated Fast Fourier-Transform (IpFFT) uses neighboring intensity value to give fractional pixel value.

Other common schemes are biased (Quinn & Hannan '01)
Phase extraction methods …

The Hilbert transform does something like this to sinusoidal frequencies:

\[ H: \sin \phi(t) + \eta \rightarrow e^{i\phi(t)} + \xi \]

This is done by taking the Fourier transform, zeroing out negative frequencies, and taking the inverse Fourier transform. The log function (or the angle function) can “unwrap” the angle function from this by

\[ \text{Im} \left[ \log \left( e^{i\phi(t)} + \xi \right) \right] \approx \phi(t) + \varepsilon \]

which can be differentiated with a variety of standard numerical methods – each of which is flawed in its own way.

\[ n.b. \text{ choices in interpolation \& differentiation yield the various digital downshift (DDS), digital upshift (DDU), adaptive downshift conversion (ADC), and spline methods.} \]

• Error isn’t zero.
• Other talk addresses the problem when this isn’t small enough.
Error Estimates for the velocity (or frequency)

Key assumptions are that the true signal has “enough” derivatives and that the oscilloscope noise is white.

\[ E_{2\sigma} \approx \sqrt{c_1^2 \sigma^2_{\text{noise}} + c_2^2 (\partial^k \phi(t))^2} \]

**Random error**, due to oscilloscope noise and time-jitter.

**Fitting error**, due to signal not fitting the model (\(k>1\) depends on the differentiation scheme chosen)

Statisticians say:

\[ \text{MSE}^2 = \text{Bias}^2 + \text{Variance} \]

It is studied in non-parametric regression, functional data analysis, & stat. sig. processing.
A comparison of several methods ...

What follows are “average velocity profiles” generated with N=1000 simulations and presented with $2\sigma$ (both actual and predicted – when available).
Synthetic, logistic curve velocity profile

The spline method (grey and black) provides reliable error estimates (grey) which bound both the true error in \( v_{\text{spline}} \) and the \( 2c_{\text{spline}} \) deviation.

The very narrow deviation \( c_{\text{DSS-min}} \) (orange) which is highly desirable seem incongruent with the very wide deviation in rise-time estimates discussed in soon to be released report.

The ability to measure larger jumps in velocity on the sub-nanosecond time scale is clearly demonstrated for many PDV analysis methods - but perhaps not IpFFT.
PDV Uncertainty Estimation

ADC has little bias, but large deviation shows it can have least accuracy.

IpFFT exhibits a non-physical oscillations and biases the peak.

Spline method has predictive error bars validated by the smaller actual error.

ADC has little bias, but large deviation shows it can have least accuracy.

IpFFT exhibits a non-physical oscillations and biases the peak.

Spline method has predictive error bars validated by the smaller actual error.
Summary from the pending comparison report

- Hilbert-Transform/Spline methods have *a posteriori* (based on the data) error bar estimates which were statistically verified.

- Sub-nanosecond rise time (10% to 90%) in very simple examples can be very accurately estimated by fitting the velocity profile to a logistic curve.
Questions and next steps

- Comparing least-squares Peano kernel methods and their associated error bar calculations.

- What other (experimental, statistical) errors might be accounted for in the uncertainty measurements (e.g. capture effect).

- Confidence regions for mPDV systems in the same way the errorbars provide confidence-intervals.
End
PDV Uncertainty Estimation

Note the noise level

Synthetic, logistic curve velocity profile
Phase Extraction: A few details that are left out

\[ \text{Im} \left[ \log \left( e^{i\phi(t)} + \xi \right) \right] \approx \phi(t) + \epsilon \]

1. **The average error isn’t zero** on the right side … even with filters and windows.

2. **A variety of filters and windowing functions** can be used and give better and worse results … depending on the circumstances.
A comparison of several methods ...