

Next-generation PDV analysis



Institute for Shock Physics (WSU, Pullman WA)



DCS@APS (Argonne IL)



Applied Science Laboratory (Spokane WA)

D.H. Dolan
T. Ao and N.P. Brown (Sandia National Labs)
2026 PDV workshop

- There are no shortage of ways one **might** analyze PDV data
 - Despite alternatives, short-time Fourier transforms are incredibly robust and widely used
- Typical results are spectrograms and/or histories
 - Each has their merits and shortcomings
- **Is there a demand for something new?**
 - What are the essential features?
 - Distinguish must, should, or might be preserved/added
 - What platforms/workflows need to be supported?
- Consider SIRHEN as an example, not necessarily the solution

Sandia InfraRed HeterodynE aNalysis (SIRHEN)



- Officially released and documented in 2010
 - Licensed for **public** release
- Runs on any platform in MATLAB, can be compiled to executable
- Explicitly distinguishes spectrogram and history parameters
 - Avoids massive (giga- to tera-pixel) image creation
 - Supports fine-scale history generation
- Completely rewritten in version 2
 - Enhanced region of interest selection
 - Batch programming via PDV object class
 - Supports parallel processing (as available)
 - Lots of other enhancements

SIRHEN2 is far from perfect

- It shares all STFT limitations
 - Shocks are smeared over analysis duration
 - Overlapping spectral features are a problem
 - Uncertainty estimates are a work in progress
- Graphical interface has issues
 - In fairness, MATLAB graphics radically changed in release 2025a
 - Transition from Java to JavaScript is not seamless
 - Program works but interface has a few unresolved glitches
- It was never formally documented and might never be
 - Version 1 manual covers much, but not all, of the theory
 - Graphical/programmatic interface is entirely different

What **must** be preserved/added?

- General applicability, i.e. not tuned for a specific application
- Ability to read many different signal formats
 - Digitizer binaries (Keysight, Lecroy, Tektronix, ...)
 - Laboratory formats (DIG, SDA/TAF?)
- Moderately quick yet adjustable spectrograms (MB to <GB)
 - Initial assessment of experiment success/failure
 - Helps crop out unneeded data (signals may contain gigapixels...)
- Some form of history extraction
 - Contextually meaningful analysis parameters
 - Varies considerably with digitizer and experiment

What **must** be preserved/added?

- More accessible
 - Compilation...
- Interactive ROI selection
- Documentation
- Scriptability

What **should** be preserved/added?

- Improved zero padding control for frequency interpolation
 - This matters a lot more than widely realized
- Complex (real/imaginary) spectrum analysis
 - Power spectrum discards information, generates cross terms
- Complicated spectrum analysis
 - Multiple peaks, acceleration/chirp, time discontinuities?

What **should** be preserved/added?

- Instrumental control (bookkeeping)
 - Time corrections
- Quick analysis versus in depth analysis
 - Executable
 - Different analysis domains? (shock, ejecta,
- Repeatability and bias
 - Automation
- Took at time series with spectrogram
- Signal synthesizer

What **might** be preserved/added?

- Improved region of interest selection
 - This can be very important but subjective and time consuming
 - Can automation replace subject matter expertise?
 - Regions of disinterest, i.e. masking
- STFT alternatives?
 - Non-discrete velocities

What **might** be preserved/added?

- Non-discrete velocities
- Variable-time analysis parameters

Environment

- Programming language
 - MATLAB has a lot of advantages, but not everyone agrees
 - **Cross-platform programs are tougher than they seem**
 - "Free" software often not always free (e.g., Anaconda)
 - Many PDV users cannot rely on perpetual Internet access (Julia)
- To compile or not to compile
 - Performance benefit is *probably* marginal
 - Updates and bug fixes are harder to distribute for executables
 - Advanced features are harder to implement in an executable
 - However, a standalone binary is handy when code is fixed
 - 3+ platforms to deal with (Windows, Mac, Linux, web?, mobile?)

Summary

- There are PDV analysis codes that work to varying degrees
 - SIRHEN2 is an example that anyone can access
 - <https://github.com/sandialabs/SMASHtoolbox>
- There is plenty of room for improvement
- **What does the community want/need?**
 - No committed effort yet, but ideas/suggestions are welcome
 - Contact Tom Ao and Nathan Brown about SIRHEN2
 - Contact Dan Dolan or Tom Ao about other possibilities