DOE/NV/25946--1352

Gemini Series Experiment Data Reduction and Storage Techniques

Robert A. (Bob) Berglin

Senior Engineer, NSTec

berglira@nv.doe.gov

Presentation To The Sixth Annual PDV Workshop

November 4, 2011

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

Topics

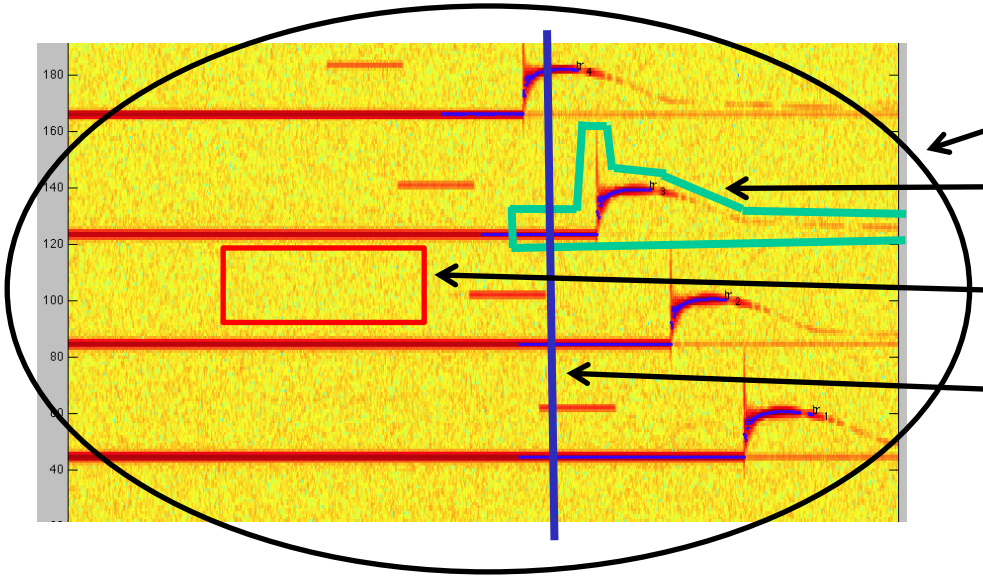
- Data Formats Expected from Gemini Experiments
- Data Quick Look versus In-Depth Analysis
- iPDV Object-Oriented Data Storage
- iPDV's Traceability of Analysis Results
- Optimizing Object Memory Usage in iPDV
- Long-Term Archival of Data Objects by iPDV
- Comments and Questions

Data Formats Expected from Gemini Experiments

- Initially Recorded Data
  - Raw digitizer data is in “P-14 Dig” file format
    - P-14 dig format is a common format for experiment data recorded at NNSS
      - Contains date/time information about the recorded event
      - Contains digitizer settings (digitizer model, horizontal/vertical settings)
    - Does **NOT** contain probe information (probe id, light wavelength, etc.)
  - Raw digitizer files are usually augmented with spreadsheet data
    - Spreadsheets have a fairly consistent format
    - Spreadsheets are formatted to be human-readable
- Storage/Interpretation of Initial Data in Matlab
  - Need to be able to read P-14 Dig files (capability already exists)
  - Need to be able to read and interpret spreadsheet files
    - Matlab has rudimentary spreadsheet support, but it will not read spreadsheet formats useful to humans
    - Matlab can be extended using external Java packages, e.g., Apache POI, to read more elaborate, human-usable spreadsheets
    - Easiest to read exported spreadsheet data in Matlab (CSV or delimited text)

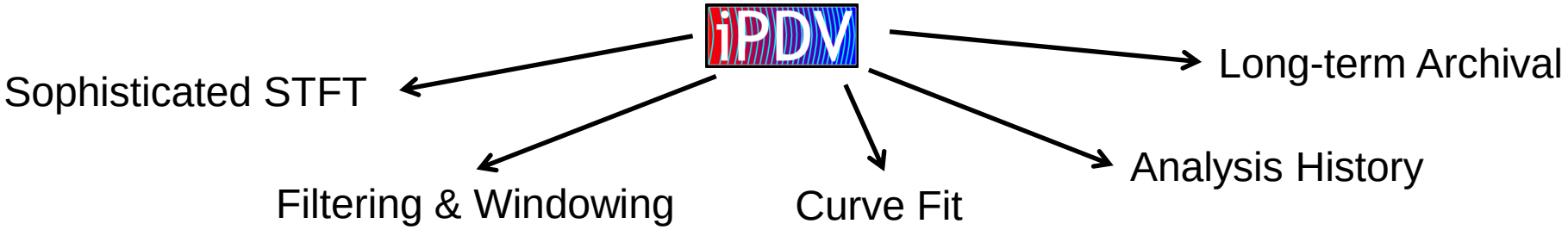
Data Quick Look versus In-Depth Analysis

QuickView – manipulates data channels for quick-look analysis

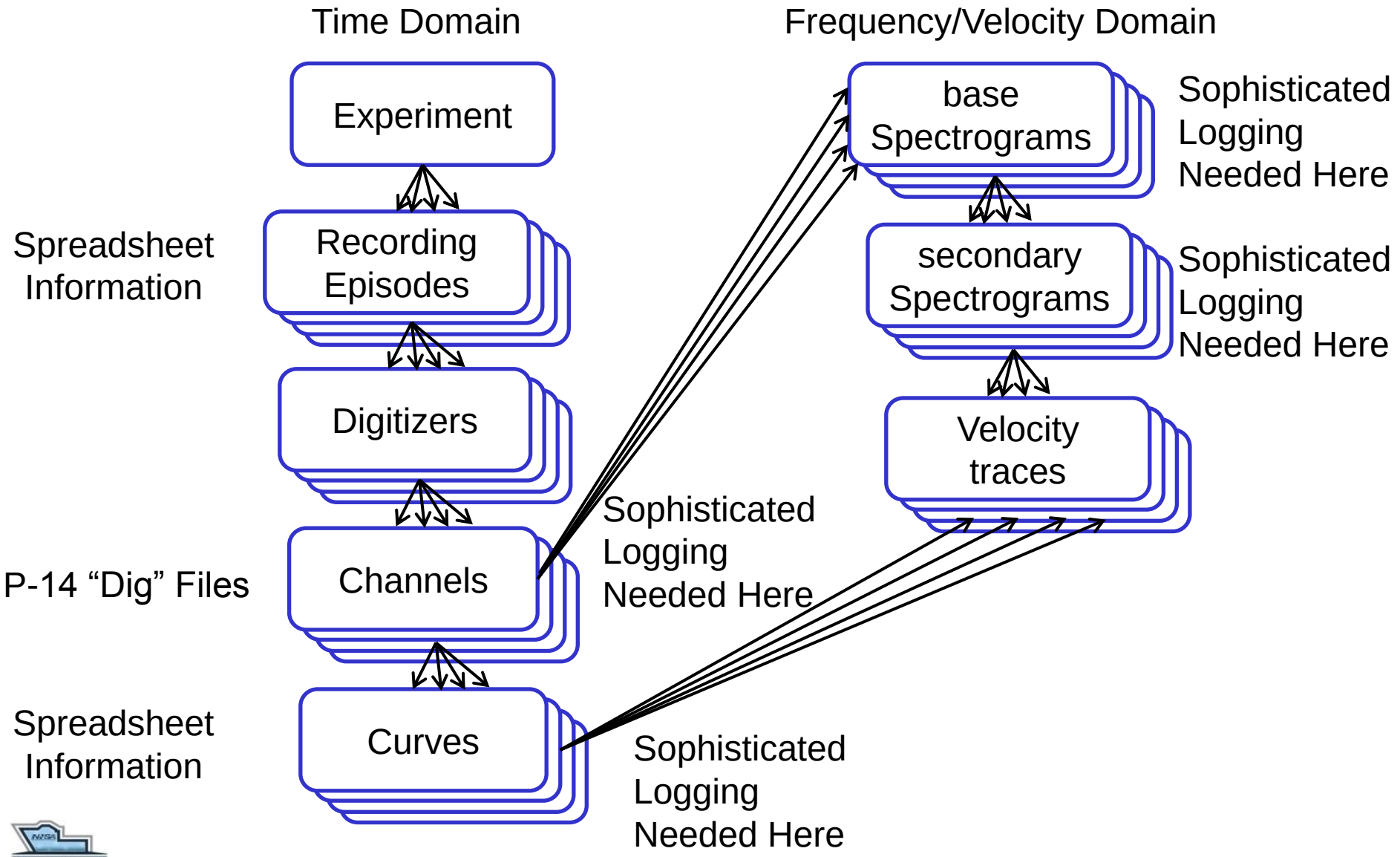


- Initial Spectrogram Creation
- Trace Delineation & Extraction
- Noise Level Estimation
- Baseline Estimation
- Initial Velocity Extraction

iPDV – manipulates individual data traces for in-depth analysis



iPDV Object-Oriented Data Storage



iPDV's Traceability of Analysis Results

- Analysis results must be traceable both forward, e.g., from a specific digitizer channel to a spectrogram, and backward, e.g., from a velocity trace back to a digitizer channel
- Some traceability is simple, such as the Experiment, Episode, and Digitizer to which a Channel belongs, or the Curve to which a Velocity trace belongs
- Other traceability is more complex, such as the traceability of the set of Spectrograms belonging to a specific digitizer Channel
  - On initial information load for an experiment, the number of Spectrograms generated for the Channel is not known
  - The parameters (metadata) used to generate each Spectrogram also need to be logged
- Traceability of Curves to Velocity traces needs to include information on the intermediate objects (the Spectrograms), and some metadata (extraction polygons, noise polygons, etc.)

Optimizing Object Memory Usage in iPDV

- Object construction and log organization need to be done in a way that minimizes object memory usage
  - Data processing, particularly spectrogram generation, requires large amounts of system memory
  - Some objects, e.g., Channels and Spectrograms, carry a lot of data that may not need to be loaded for some processing operations
    - Separation of data and metadata load preferred
    - Separation of these load operations is mildly difficult using internal Matlab object storage (.mat files)
- Log information memory usage is a lesser consideration, but may be amenable to a reduced memory footprint organization as well

Long-Term Archival of Data Objects by iPDV

- Matlab-style file storage (.mat files) is not a good choice for long-term PDV data storage
  - Proprietary format
  - Subject to vendor version changes, vendor viability, or changes in user-community product choices
- Matlab has minimal support for other data file formats:
  - CDF and netCDF
  - FITS
  - HDF4 and HDF5
  - Formatted text
- iPDV will use HDF5 format for long-term storage of data
  - HDF5 is an open-source format
  - HDF5 is supported by the DOE Advanced Strategic Computing Initiative (ASCI) Program
- iPDV data objects will be stored in per-object HDF5 files
  - Much of the metadata stored as HDF5 attributes in the files
  - Viewable using non-Matlab tools (e.g., HDFView)

Comments and Questions

Comments? Questions?