The SonATA DX

Ken Smolek

Revision 0.04, 2008 January 11

The DX performs all creation of channels, spectrometry and signal detection from the channel level down. A channel is equivalent to a narrow-band PDM; each channel is processed independently of the others. Two designs are considered:

1. A single executable of the DX software will process several channels. Each channel will have its own buffers and state and be processed independently of the other channels, except for signal detection, which will be done serially by a single task using a single large detection buffer.
   Advantages:
   1. Scheduling efficiency may be better.
   2. Less memory is used, particularly for very long observations.

   Disadvantages:
   1. More complicated because several independent channels must be kept separate.
2. The DX software will process a single channel. A host DX (the physical computer) can process multiple channels by simultaneously running multiple copies of the DX sofware. Channel processing will then be completely independent, including signal processing.
   Advantages:
   1. Simpler architecture.

   Disadvantages:
   1. Uses more memory for the same number of channels.
   2. Process switch overhead may degrade performance slightly.

Timing Constraints

A two-stage pipeline is assumed:

1. Data collection:
   1. Subchannelization.
   2. Spectrometry.
2. Signal detection:
   1. CW power detection.
   2. Pulse detection.
   3. Coherent detection of CW power candidates.
   4. Confirmation of secondary (i.e., other-DX) candidates.
   5. Archiving.

Stage 2 must take place in a time less than or equal to that of stage 1, so that data collection for activity n+1 may begin immediately after data collection for activity n is complete. This is not essential for proper functioning of the system, but increases observing efficiency.

Subchannelization

1. Accept X & Y linear polarization channel packets from the X & Y Channelizers via 1GbE.
   1. Verify beam and channel.
   2. Samples are 16-bit integer complex (16-bits real, 16-bits imaginary).
   3. Separate X & Y packet lists are maintained. Missing packets are replaced with zero-filled packets. Late packets are discarded.

2. Convert input samples to single-precision (32-bit) floating point complex.
3. Collect statistics on X & Y input streams.
4. Convert X/Y linear polarization to L/R circular polarization.
5. Collect statistics on L & R output streams.
6. Buffer L/R data until there is enough to perform the DFB/FFT (one half frame).
7. Perform a 10-folding DFB followed by an N-point FFT to create subchannels of ~675 Hz.
    1. N is TBD, but will be between 768 and 1024 points, depending on the amount of oversampling.
    2. Due to oversampling by the Channelizers, the incoming channels are wider than the spacing between channel centers, so some outer subchannels will have to be discarded. The number of subchannels discarded at each edge will depend upon the amount of oversampling by the Channelizer and N, the number of subchannels being created.
8. Collect statistics on the subchannels.
9. Corner turn the subchannel data into a half frame buffer.
10. Notify the spectrometer that a half frame of data is ready.

Spectrometer

1. Receive a message from the subchannelizer that a half frame of data is ready. (Both polarizations.)
2. For each subchannel:
    1. Use the new data plus the current baseline to compute new L & R baselines.
    2. Apply the current baseline to the new data.
    3. Compute confirmation (CD) data by converting subchannel samples from singe-precision FP complex to 4-bit integer complex. Saturate values > 15 and < -15.
    4. Copy data to CD buffer.
    5. Call libSpectra for both L & R baselines. This will create all requested resolutions (up to 10 resolutions).
    6. For each resolution:
        1. Threshold for pulses.
            1. Add pulses to pulse vector.
        2. If DADD resolution:
            1. Compute DADD power (2 bits) for each bin, pack 4 bins/byte.
            2. Copy data to CW buffer.
3. Report baseline data if required.

Signal Detection

1. CW power detection (DADD):
    1. Process L & R polarizations independently.
    2. Process entire channel as a single 400KHz slice.
    3. Unpack 2-bit power data into 1-byte accumulators.
    4. Threshold pathsums and report hits found.
2. Pulse detection:
    1. Build a pulse map from the pulse vector, combining polarizations.
        1. Map is indexed by resolution/bin/spectrum.
    2. For each resolution:

        1. Search for and report pulse trains. Processing is performed within overlapping slices.
        2. Search for and report large singleton pulses.
3. Cluster signals.
4. Classify signals.
    1. Reject zero drift.
    2. Reject RFI.
5. Report signals to SSE.
6. Do coherent refinement of CW candidates and report results.
7. Archive candidates (both CW and pulse) as requested by SSE.

Confirmation

1. Uses confirmation (CD) data to synthesize frequency range of interest.
2. Primary coherent detection:
    1. Performed on all CW candidates found by the CW power detection.
    2. Provides a more precise description of coherent signals.
    3. Report results to SSE.
3. Secondary coherent detection:
    1. Performed on candidates sent by the SSE; these will generally be candidate signals found by other beams.
    2. Does limited-range search for the candidate.
    3. Report results to the SSE.
4. Secondary pulse detection:
    1. Create spectra/bins containing the pulses.
    2. Sum the powers in the corresponding pulses, then do a chi-square test for significance.
    3. Report results to the SSE.

Archiving

1. Controlled by the SSE: no candidate is archived until the SSE requests it.
2. Limited to primary and secondary candidate signals.
3. The archiver connection is separate from the SSE connection.

Possible Directions for Development

1. DADD enhancements:
    1. Perform sum-of-polarizations as well as L & R.
    2. Perform DADD on wider bins (i.e., up to 1KHz).
    3. Use larger accumulators to minimize the need for shift-normalization during path summing.
2. Dedrifted coherent detector.
3. Better clustering algorithms.
4. Consider using small-folding DFB for final spectrometry.
5. More sophisticated secondary-candidate processing.
6. Look for “doubleton” pulses, pairs of pulses which are far above pulse threshold but less than singleton threshold.