Real-Time RFI Mitigation for Single-Dish Radio Telescopes

Richard Prestage, GBO
Collaborators

• Cedric Viou, Jessica Masson
  – Station de radioastronomie de Nançay
    Observatoire de Paris, PSL Research University, CNRS,
    Université d’Orléans

• Nick Joslyn, Emily Ramey – GBO REU Students

• Tim Blattner – NIST

• Michael Lam - West Virginia University

• Luke Hawkins, Jason Ray, Mark Whitehead - GBO
Talk Outline

• Motivation and science goals
• Approach
• Time and frequency domain blanking
• Implementation and initial test results
• Next steps
MOTIVATION

• Problems caused by RFI continue to grow:
  – Increasing occupancy of RFI
  – Wider bandwidth observations
  – Ever increasing data rates
  – More sensitive telescopes

• Current approaches are becoming unsustainable

• Single dishes are more susceptible than Interferometers

• Despite all of these reasons, GBT observations continue to rely on offline, semi-interactive RFI mitigation approaches

• Goal is to provide a complete implementation for the GBT VEGAS spectrometer / pulsar backend, which may then also be used in other similar instrumentation
Approach

• Develop real-time identification and mitigation algorithms which can be implemented in the heterogenous FPGA / CPU / GPU VEGAS DSP pipeline

• Previous GBT work has raised skepticism about “black-box” implementations, and concerns about unknown impacts on data quality

• Prototype and rigorously qualify approach using archival raw voltage data

• Work closely with domain experts to ensure validity of approach at the level of improved astrophysical results, not just “nicer looking spectra”
Science Target I: Pulsar Timing

• Science Goals
  – Detection of gravitational waves via pulsar timing arrays
  – Precision tests of general relativity
  – Constraining neutron star equations-of-state

• Observing Mode: coherent dedispersion and real-time folding
  – RFI mitigation performed offline, on ~ 10 second accumulations

• Lam et al. 2016:
  – Template fitting errors dominate TOA precision for many [NANOGrav] pulsars for many epochs [so increasing effective bandwidth worthwhile]
  – Errors ... introduced from unremoved RFI will produce extra variance on short timescales
Science Target II: HI emission in gravitationally lensed galaxies

- Star formation rate has plummeted in last $\sim 8$ Gyr
- HI content of galaxies (via DLA) constant since $z \sim 2$
- Statistical measurements of the cosmological HI mass density (stacking, intensity mapping) consistent with DLA results
- BUT: these approaches cannot study HI content of individual galaxies

- Arecibo: $z \sim 0.25$ (Catinella et al. 2008)
- CHILES with VLA: $z \sim 0.5$
- GBT + lensed gals: $z \sim 0.7-0.8$
Test Data

- L-band observations of pulsar J1713+0747, obtained as part of a NANOGrav global timing campaign
- GUPPI “raw” complex voltage data – 200 MHz BW, 32 coarse channels
  - 6.25MHz bandwidth, 0.16 µs time resolution
- Multiple radar and tone signals
- ARSR-3 FAA Air Surveillance Radar at 1256 and 1292 MHz
  - 2 µs pulse with an average repetition rate of 341 pps
  - sweep rate of 5 rpm (12 second rotation period)
  - Normally suppressed by an RF notch filter between 1.2 and 1.34 GHz
    (i.e. 140 MHz of lost bandwidth)
Example spectrogram
Example spectrogram

INFILE=/media/ceman/g/026.ldf
c job=guppi1 FC=1318.750 MHz FS= 6.250 MSPS
#CHN:4096 ( 0.655 ms x 1.526 kHz)
#FFT:16384 (10.737 s) SGR: (64.2 ms x 7.6 kHz)
Start:UTC 1970 Jan 1 00:00:00. POL=x STAGE=0 BLK:1

freq [MHz]

0 0.5 1 1.5 2
1316 1317 1318 1319 1320 1321

time [s]

0 0.9 1 1.1 1.15 1.2 1.25
0 2 4 6 8 10
Approach

- Mitigate impulsive broadband RFI using time-domain blanking in FPGA
  - Robust Recursive Power estimator
  - Strong and weak Bernoulli outlier detectors
  - Low computational complexity appropriate for FPGA implementations

- Mitigation narrowband RFI using frequency-domain blanking in CPU/GPU
  - Perform forward FFT with time and frequency resolution matched to expected (or automatically learned) characteristics of the RFI present
  - Accumulate as necessary to increase INR
  - Identify outliers using Median Absolute Deviation (MAD) on power spectra
  - Flag affected channels in non-accumulated data; IFFT
  - Process cleaned time-domain voltages through real-time pipeline, as before
  - High computational complexity requires CPU / GPU implementation

[Some results shown at end use MAD in time domain also]
Time Domain Blanking

- Uses the instantaneous power from complex voltages (i.e., from a PFB) as detection criterion => 2 mult + 1 Acc => cheap to implement.
- RFI occurrence is decided when the instantaneous power deviates over a threshold for a chosen period of time related to the RFI pulse length.
- Since the instantaneous power of a centered gaussian random distribution follow a chi² distribution, only the estimation of the mean power is needed to fully “know” the distribution => no need for a costly and uncertain subsequent estimation of variance to compute a detection threshold.
- The threshold is solely based on the mean power estimation that is implemented using a recursive low-pass filter.
Time Domain Blanking

• Since detection is quick, we can prevent the mean power estimator from using corrupted samples, leading to a Robust Recursive Power (RRP) estimator.

• This only adds very little hardware (a Mux) to the classical recursive mean power estimator compared to other implementations using FPGA-implemented MAD estimators.

• The detector can be easily extended ($z^{-1}$ delays replaced by $z^{-nb\_chan}$) to process independent interleaved channels that are naturally present at the outputs of PFBs provided by the CASPER library.
RRP Estimator

- RFI detection flag from thresholding module

- Multiple delays for RP estimation of several channels
Strong and weak pulse detectors

3 of 3 samples
> strong threshold

25 of 30 samples
> weak threshold
Data Replacement

- Replace corrupted samples by clean samples previously recorded in a Dual-Port Memory (one port for storing, the other one for fetching)
- Preserve power levels, even with interleaved channels since a correct memory mapping keeps samples separated
- Provides randomness in sample ordering for each channel using LFSR and data itself for address generation
- The latest sample read from memory is replaced as soon as possible by a new clean sample
Data Replacement

- Complex interleaved data stream with corrupted and clean samples
- RFI-free flag => OK to store
- Complex interleaved data stream with clean samples only
Example – air traffic radar
Input stream power
Output stream power
RRP output
Strong pulse threshold = 4 x RRP
Weak pulse threshold = 0.9 x RRP
Frequency Domain Mitigation

• FFT a series of N x M time samples
  – Append complex frequencies to N x M AppendBuffer
  – Accumulate N power spectra to single M-point IntegrationBuffer
• Apply MAD algorithm to IntegrationBuffer
• Replace complex values at corresponding frequencies in AppendBuffer
• Inverse FFT and proceed with original processing
Frequency Domain Mitigation
Frequency Domain Mitigation

![X Power Spectrum After RFI Mitigating](image)

![Y Power Spectrum After RFI Mitigating](image)
Pulsar TOA Residual Results

Chan 23 unmitigated Ave TOA: 23.73 +/- 488.42 us

Chan 23 zapped Ave TOA: 3.74 +/- 107.73 us

Chan 23 mitigated Ave TOA: 1.22 +/- 4.40 us
Pulsar TOA residual results

Channel 19 unmitigated Ave TOA: 8.33 +/- 4.23 us

Channel 23 unmitigated Ave TOA: 23.73 +/- 488.42 us

Channel 23 mitigated Ave TOA: 1.22 +/- 4.40 us

Time (s)
Hybrid Task Graph Scheduler (HTGS)

• Time-domain RFI mitigation (and the remainder of the CPU / GPU DSP pipeline) is complex
  – We wish to precisely define and document the algorithms and processing stages
  – Significant interaction / overlap between I/O and computation, memory management and task scheduling
  – Wish to optimize the design to maximize throughput and hardware utilization

• HTGS approach provides considerable assistance:
  – graph representation from the model and framework is explicit
  – provide a separation of concerns between computation, state maintenance, memory, and scalability
  – allows rapid prototyping and experimentation for performance
Hybrid Task Graph Scheduler (HTGS)

- Development to date:
  - Prototyped initial data access tasks and computational stages in Python.
  - Ported to naïve C++ implementation
  - Developed initial HTGS task graph design
  - Create a `htgs::ITask` for each computational entity
  - `htgs::IData` is used to represent data required by each `htgs::Itask`
  - Fill out HTGS design using initial C++ code

- HTGS version provided 26x speed improvement compared to initial vanilla C++
  - 4 cores, 2 threads
  - 8 Thread implementation
Summary

- We have defined an end-to-end real-time RFI mitigation approach for single-dish spectrometer / pulsar backends, utilizing both time and frequency-domain mitigation.
- Initial offline prototypes have been developed utilizing Python and Simulink, and tested using archival GUPPI raw voltage data.
- Results are / will be evaluated utilizing rigorous astrophysical metrics (pulsar TOA analysis underway; redshifted HI spectrum analysis soon).
Next Steps

• Complete and test Roach-II time-domain blanking implementation, including configuration and control options

• Complete and test HTGS frequency-domain blanking implementation, including partitioning between CPU / GPU

• Commission using multiple VEGAS Banks, receiving identical copies of the same IF signal, one utilizing blanking, one without
The Green Bank Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.