Simulation of Burst Waveforms and Burst Event Triggers

Alan Weinstein
Caltech
Burst UL WG
LSC meeting,
8/13/01

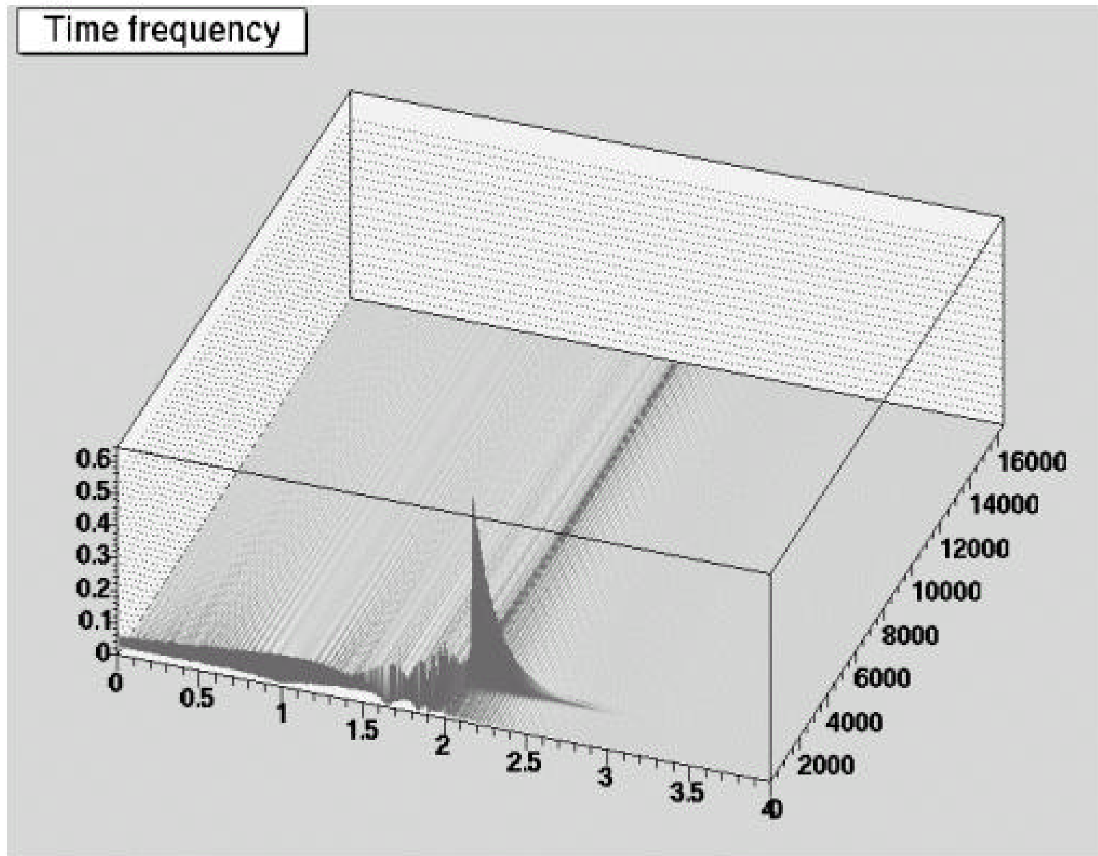
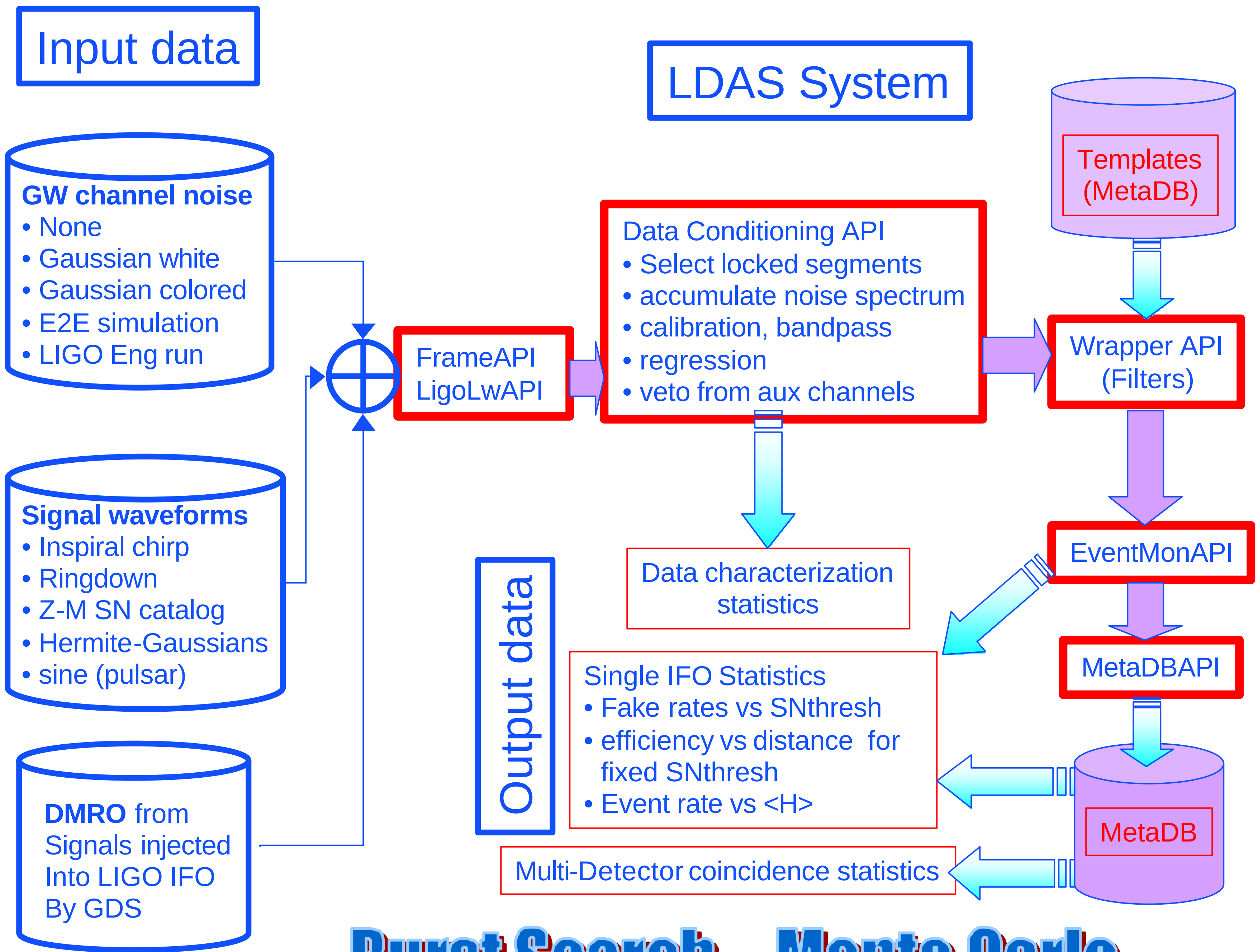


Fig. 3 Spectrogram of a composite signal

Models for unmodeled astrophysical waveforms

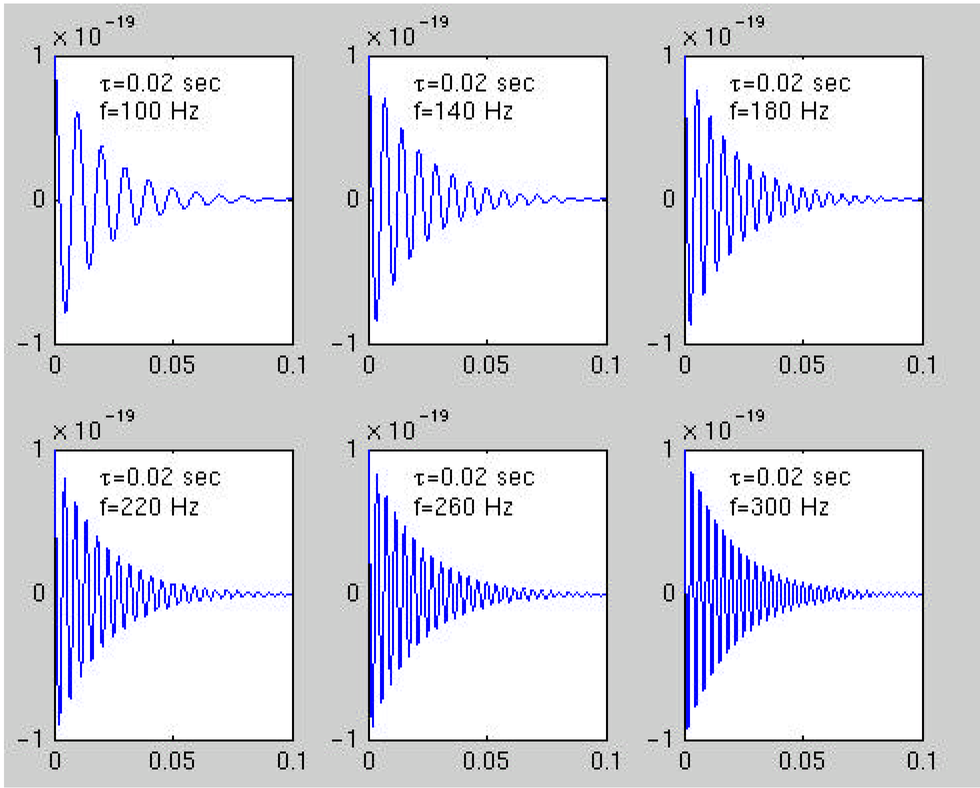
- MATLAB code has been prepared to generate frames with any combination of:
  - Signal waveforms
  - Chirps, ringdowns, Hermite-Gaussians, Z-M S/N waveforms
  - Noise: none, white, colored gaussian (***simData***), E2, E5
- Including effects of:
  - Detector calibration / frequency response (E2 only, so far)
  - Detector antenna pattern (if desired; but we don't)
  - Delays between IFOs
  - Resampled/decimated to any ADC rate (16384, 2048, …)



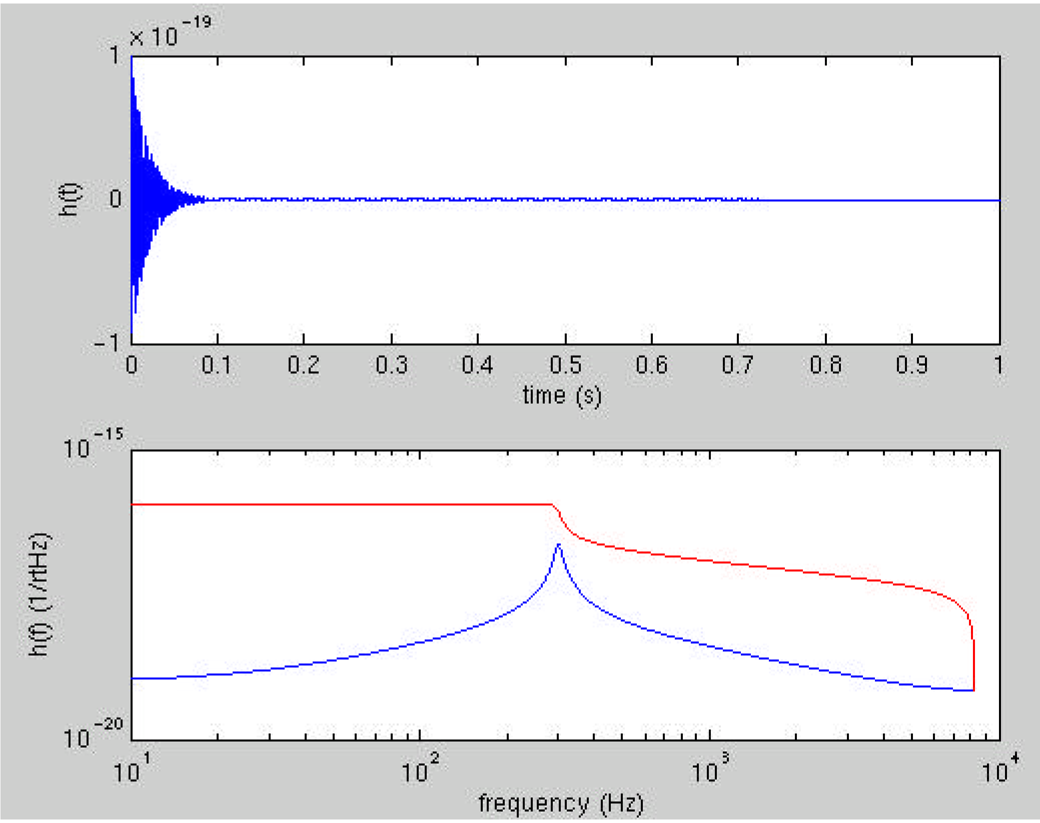
Burst Search – Monte Carlo

Ringdowns

- Rationale:
  - » Just a way to represent a burst with limited duration, abrupt rise and gradual fall, with some wiggles.
  - » Very well-defined peaked PSD
- Parameters:
  - » Peak h
  - » Decay time τ
  - » Ring frequency f_{ring}

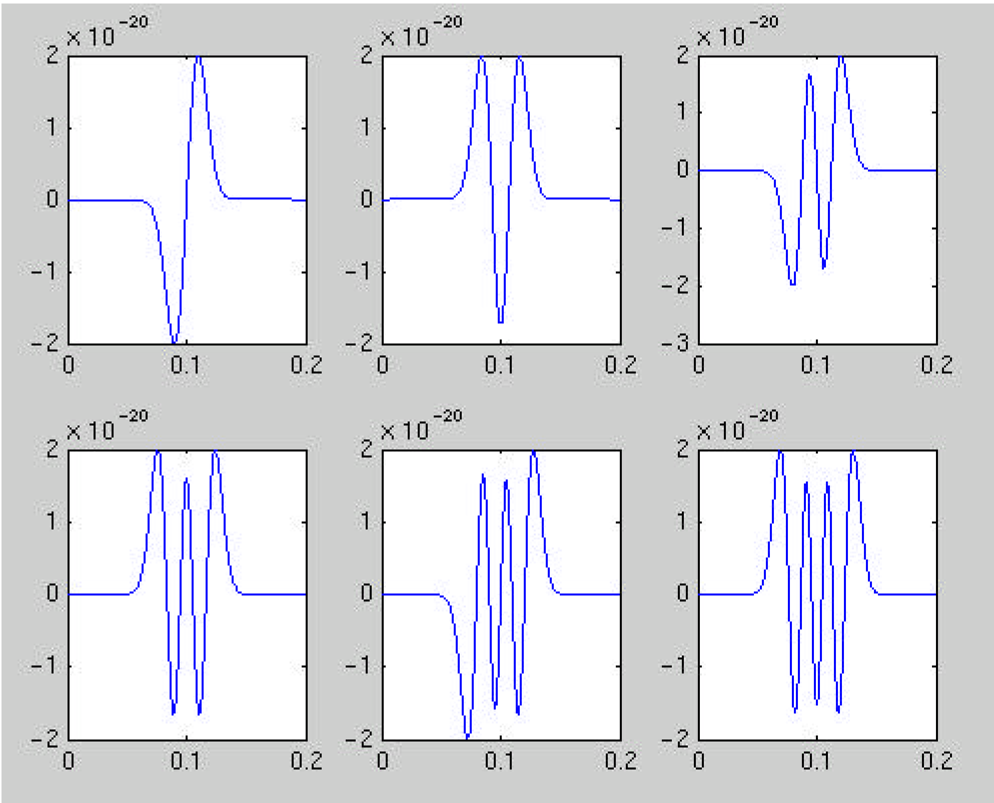


Ringdown PSD

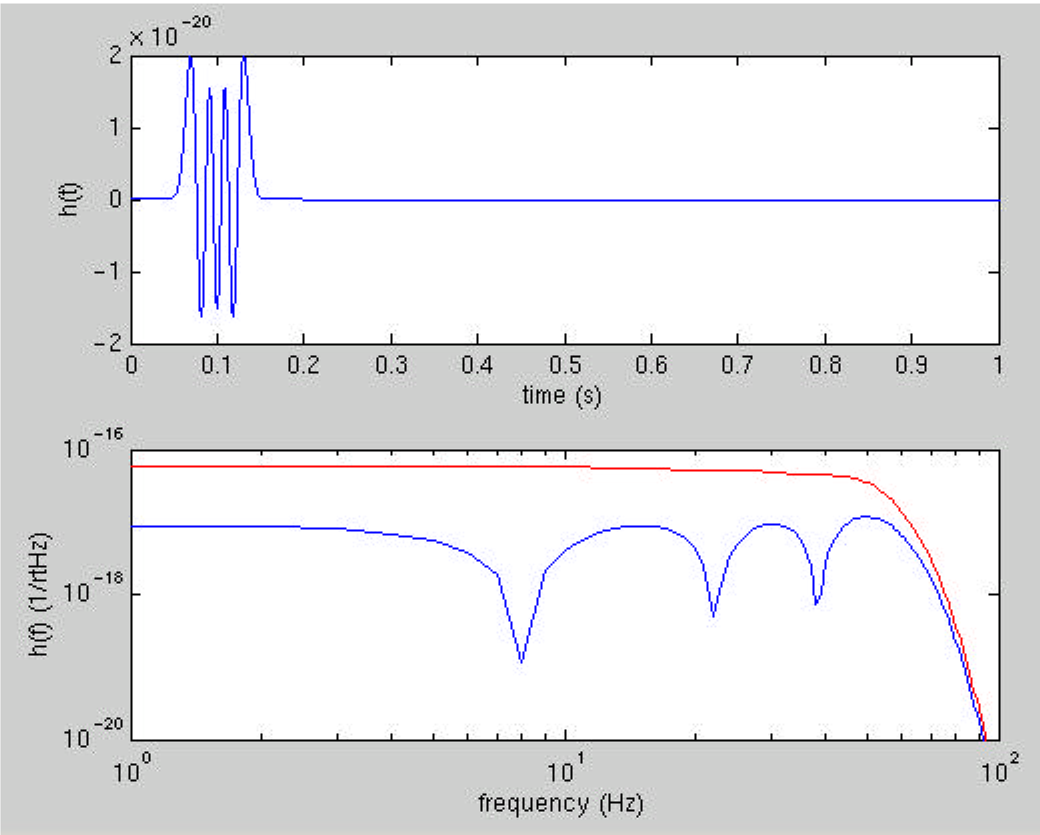


Hermite-Gaussians

- Rationale:
  - » Just a way to represent a burst with limited duration, gradual rise and fall, with some wiggles.
  - » Can also do sine-gaussians, etc
  - » Many beats in the PSD
- Parameters:
  - » Peak h
  - » Gaussian width in time, τ
  - » Hermite order (number of wiggles)

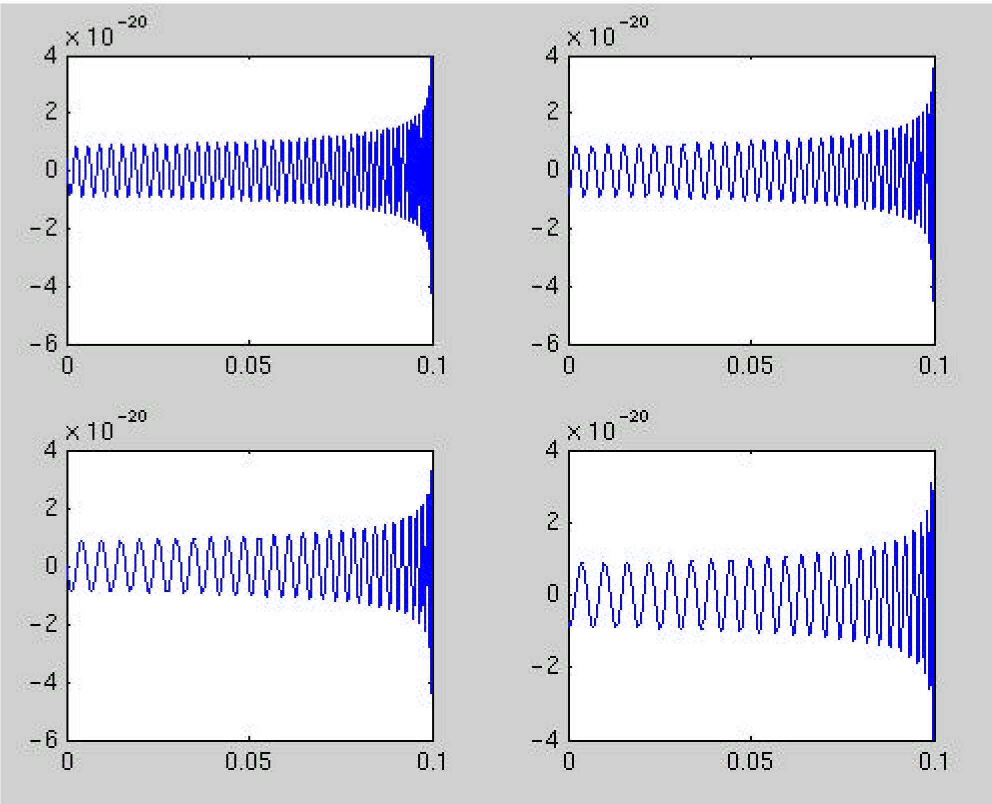


Hermite-Gauusian (6th order) PSD

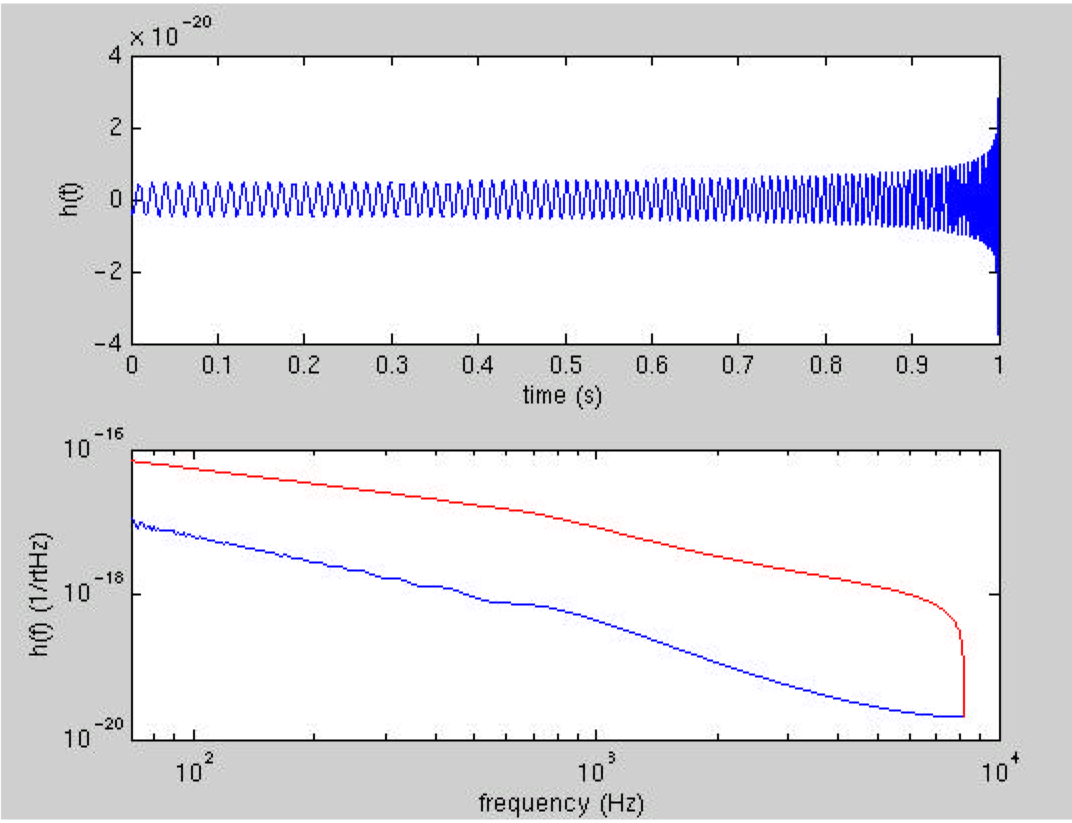


Chirps

- Rationale:
  - » In case the inspiral filters are not operational for some reason…
  - » Just a way to represent a burst with limited duration, gradual rise and abrupt fall, with wiggles.
  - » Well-defined power-law PSD
  - » Simplest Newtonian form; not critical to get phase evolution right since we're not doing matched filtering
- Parameters:
  - » Peak h (or distance D)
  - » Duration Δt
  - » $f(-\Delta t)$, or chirp mass M

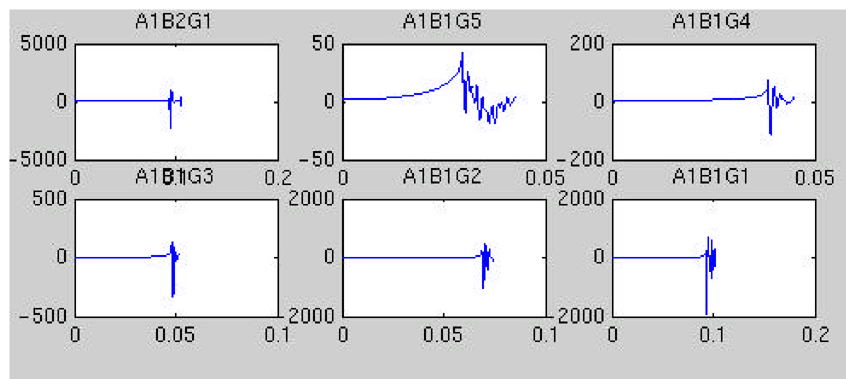


Chirp PSD

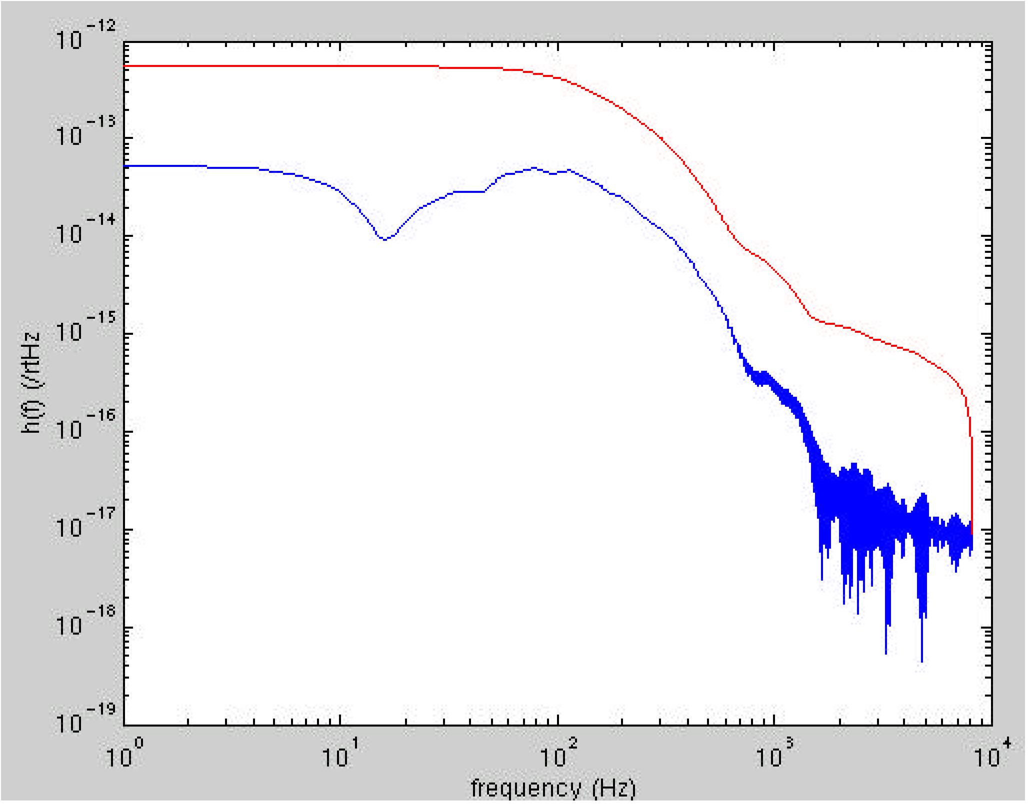


Zwerger-Müller SN waveforms

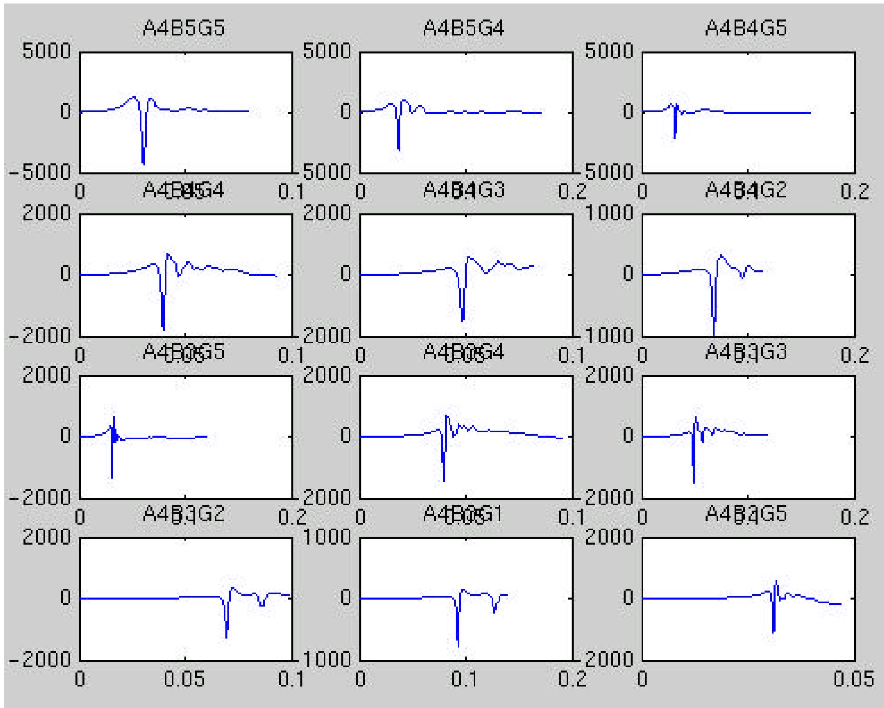
- Rationale:
  - » http://www.mpa-garching.mpg.de/Hydro/GRAV/grav1.html
  - » These are "real", astrophysically-motivated waveforms, computed from detailed simulations of axi-symmetric SN core collapses.
  - » There are only 78 waveforms computed.
  - » Work is in progress to get many more, including relativistic effects, etc.
  - » These waveforms are a "menagerie", revealing only crude systematic regularities. They are wholly inappropriate for matched filtering or other model-dependent approaches.
  - » Their main utility is to provide a set of signals that one could use to compare the efficacy of different filtering techniques.
- Parameters:
  - » Distance D
  - » Signals have an absolute normalization

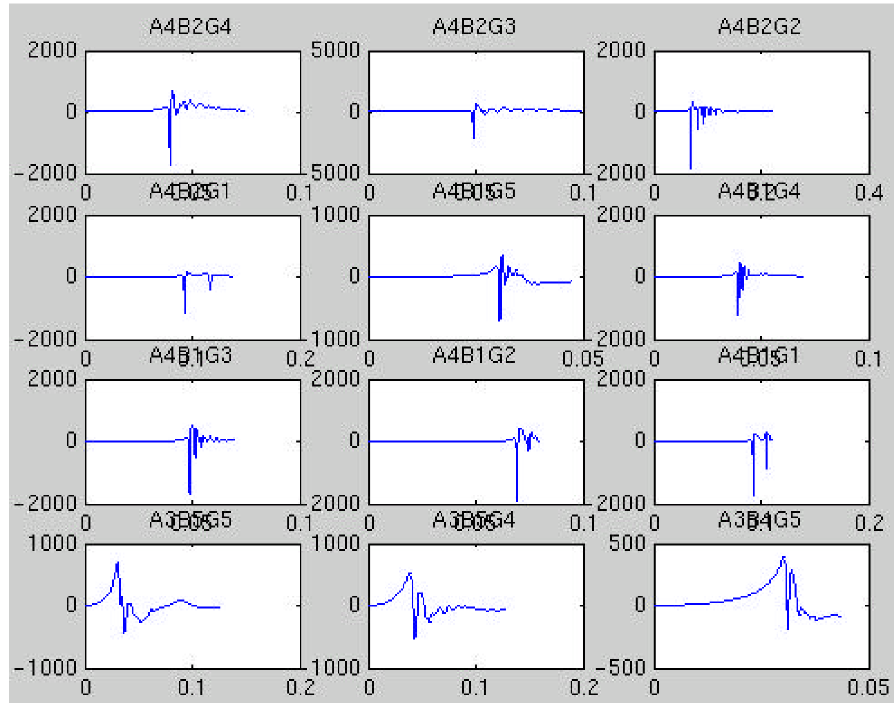


“Typical” ZM SN waveform PSD, 1 kpc

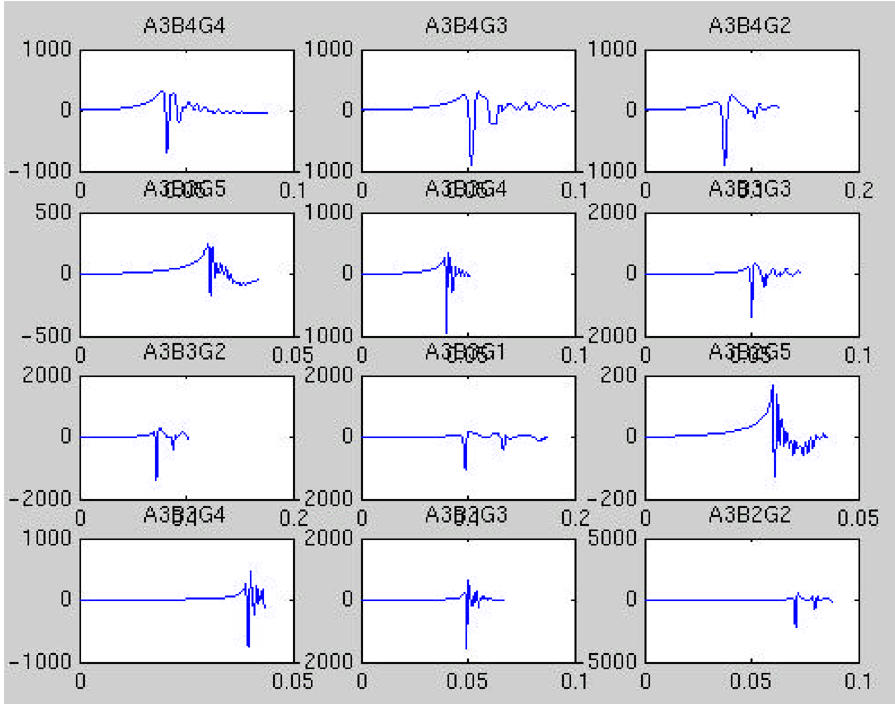


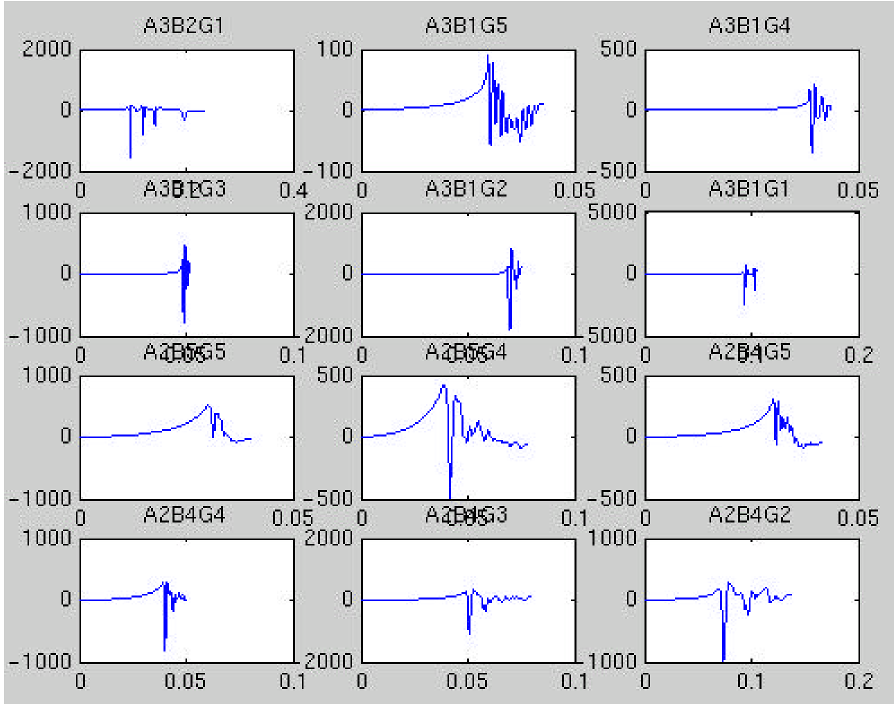
Z-M waveforms (un-normalized)



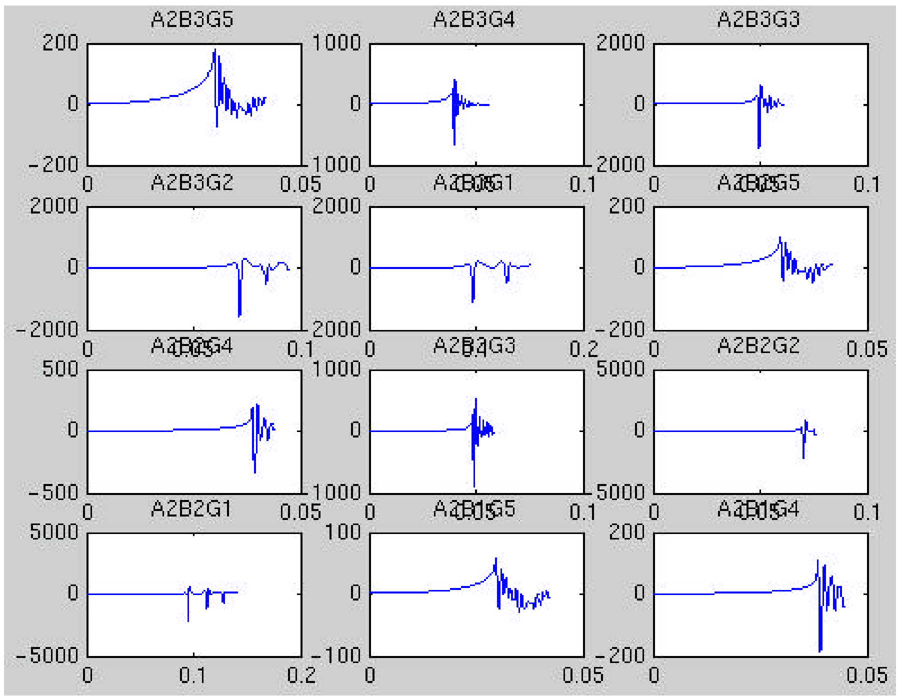


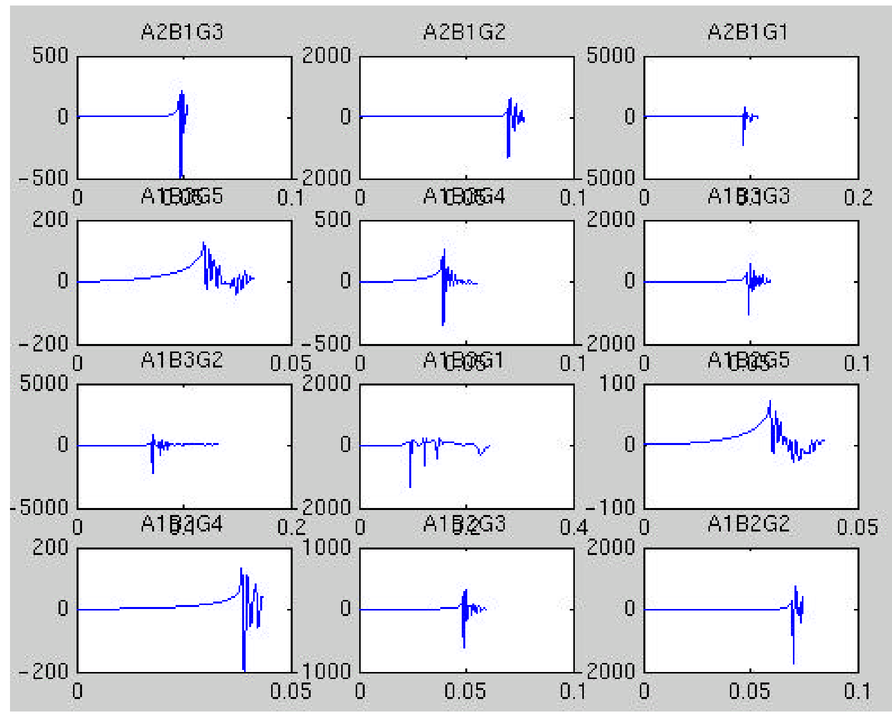
Z-M waveforms (un-normalized)



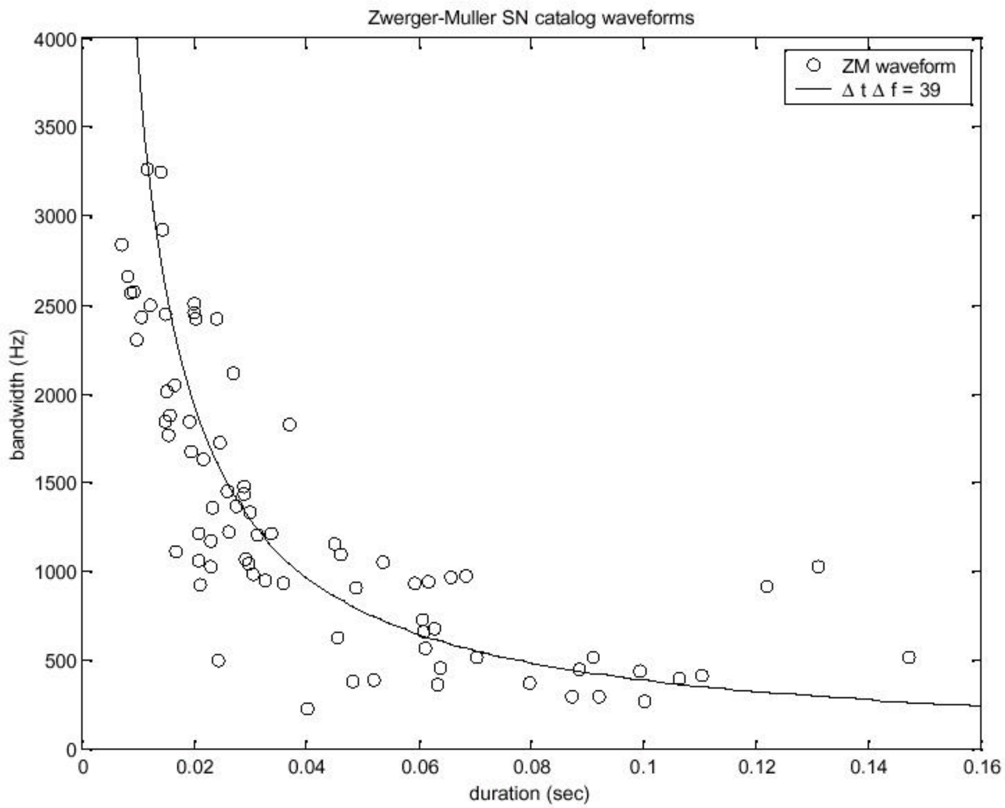


Z-M waveforms (un-normalized)

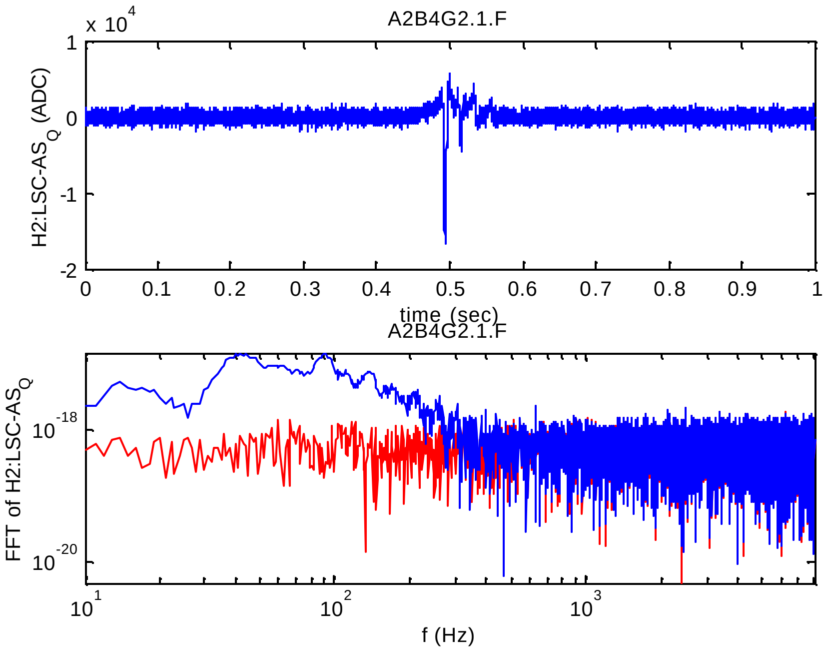


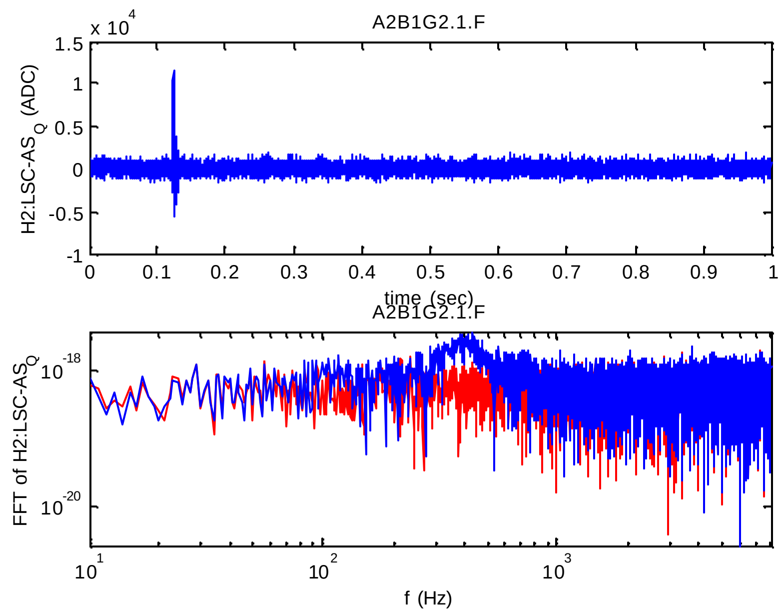


ZM waveform duration vs bandwidth

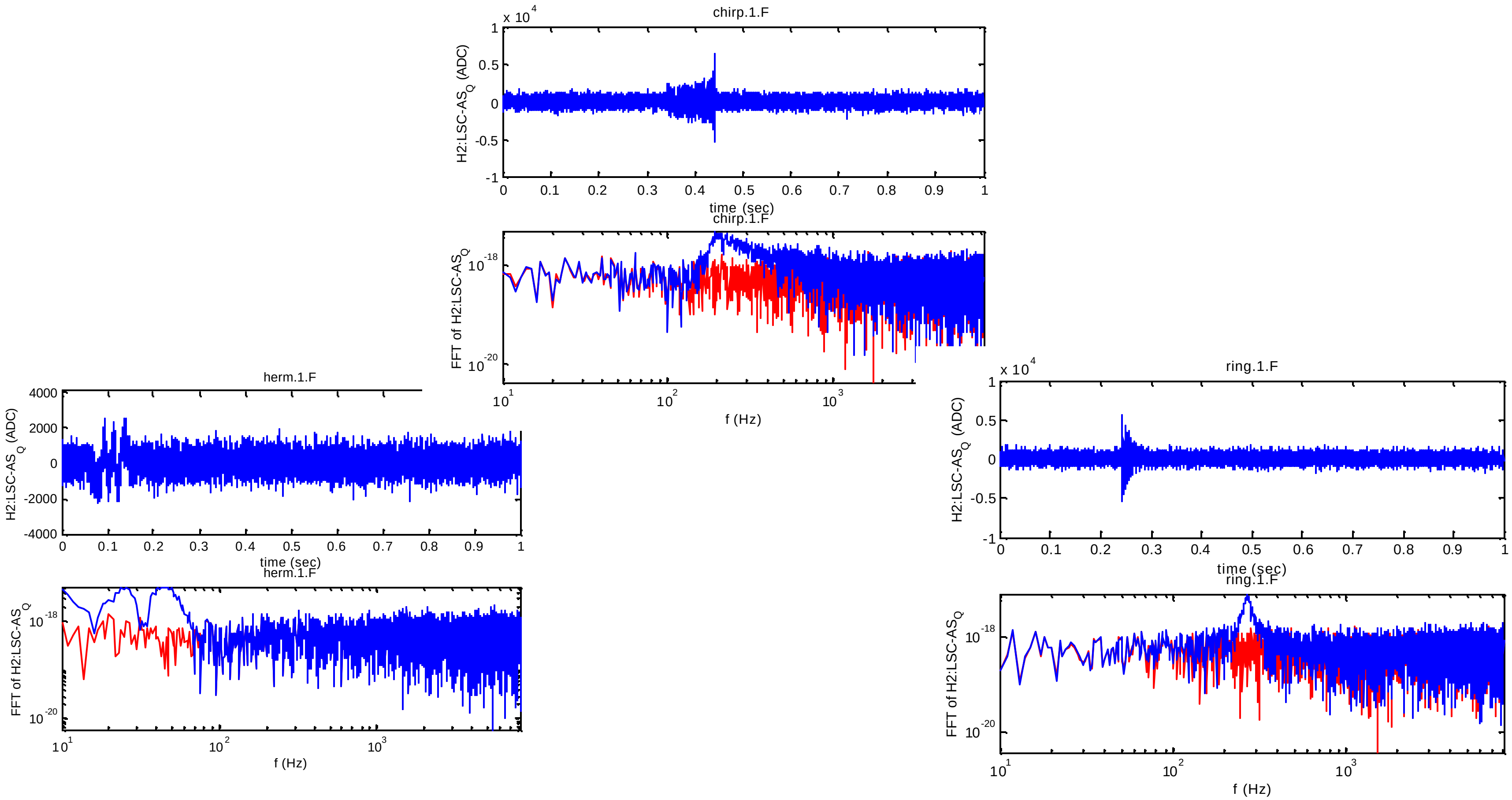


ZM waveforms buried in white noise

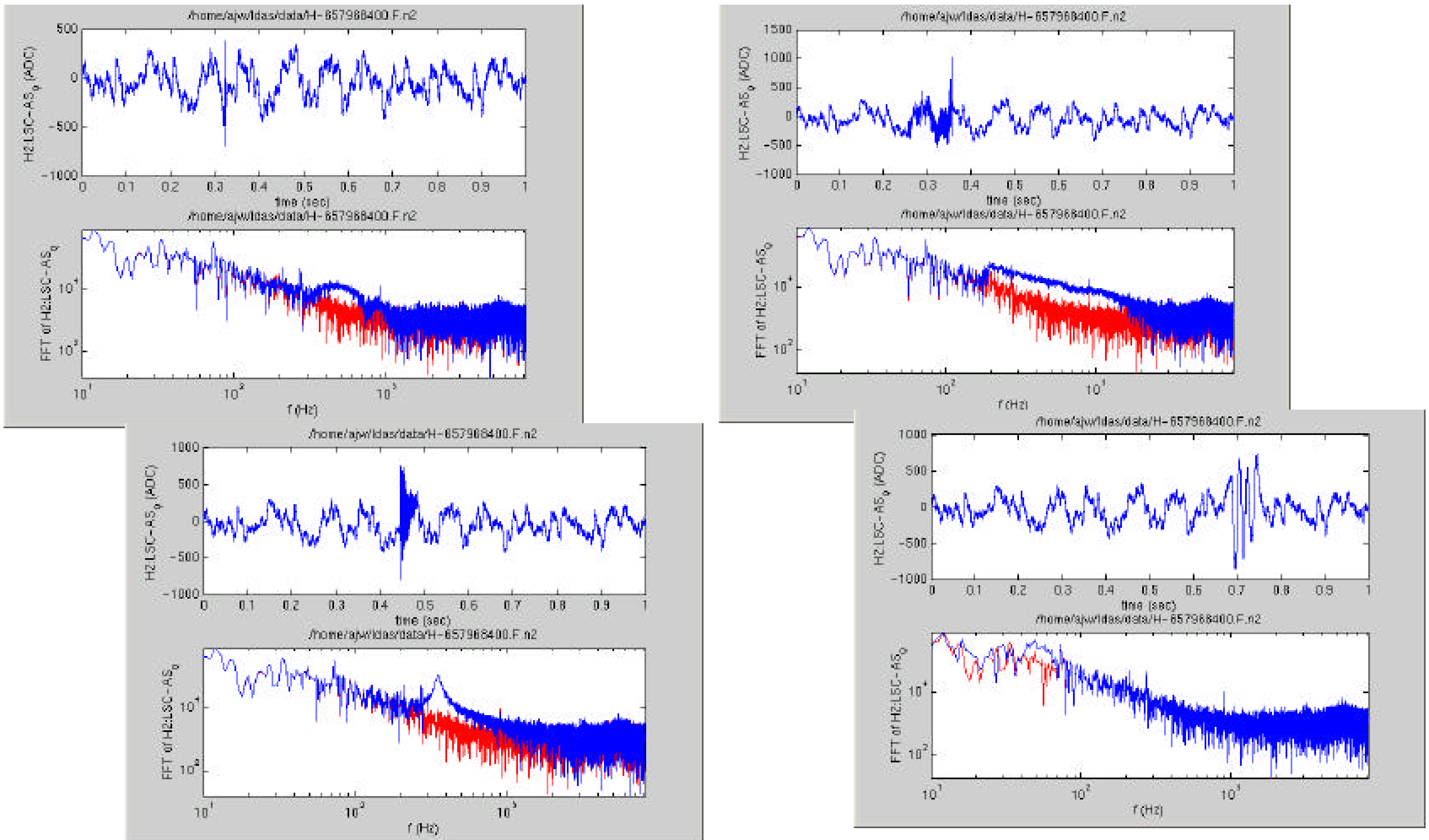




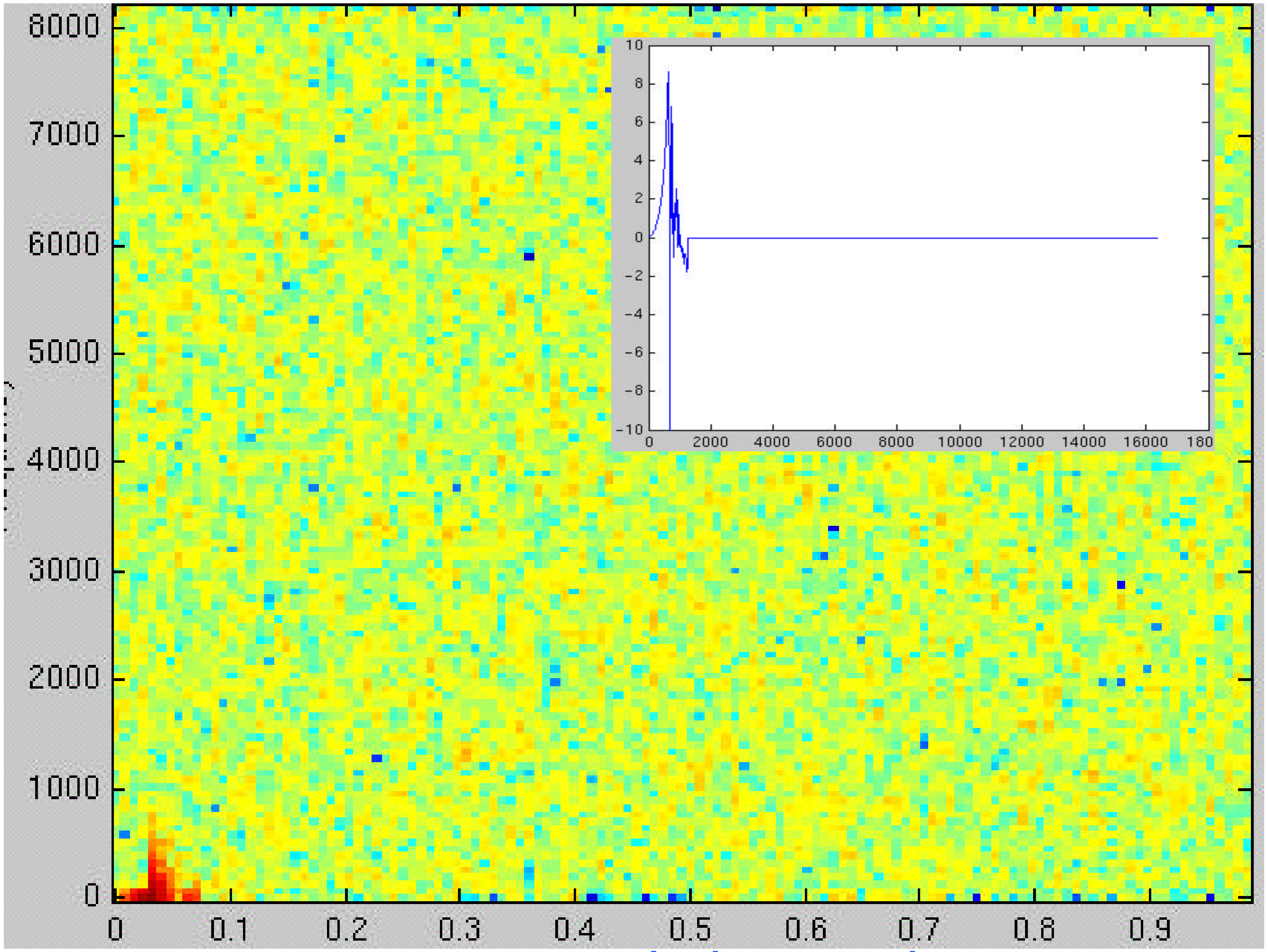
H-G, chirps, and ringdowns buried in white noise



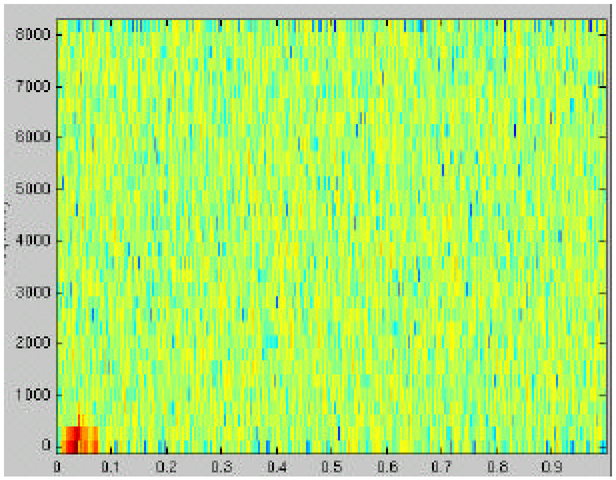
Waveforms buried in E2 noise, including calibration/TF



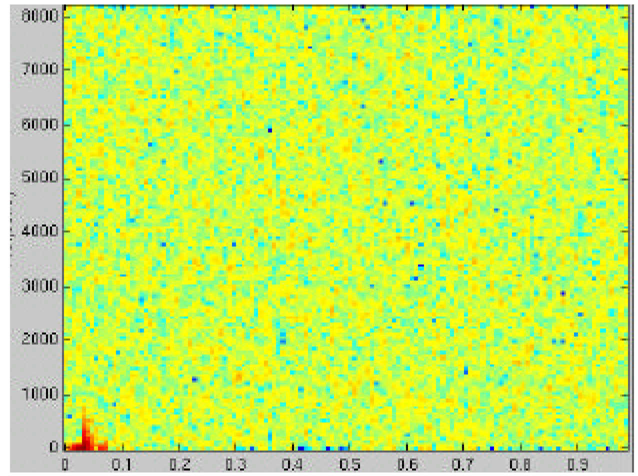
T/f specgram of ZM signal + white noise



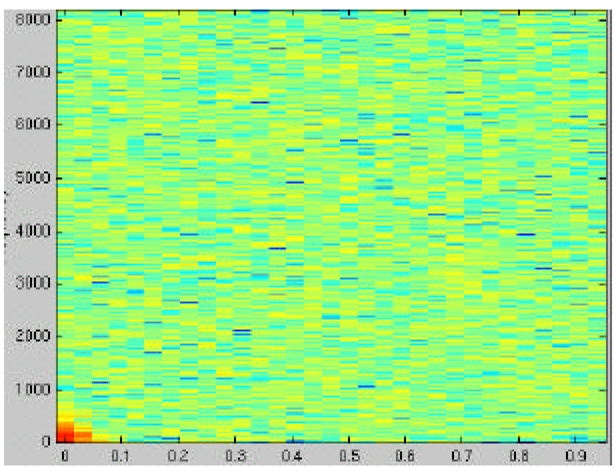
Same signal, same noise, different tf binning



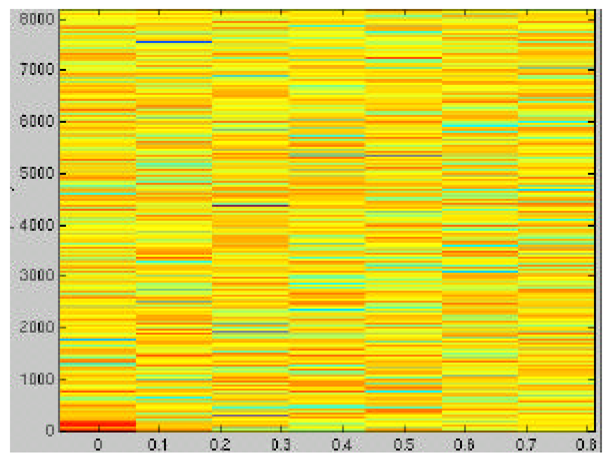
64



256

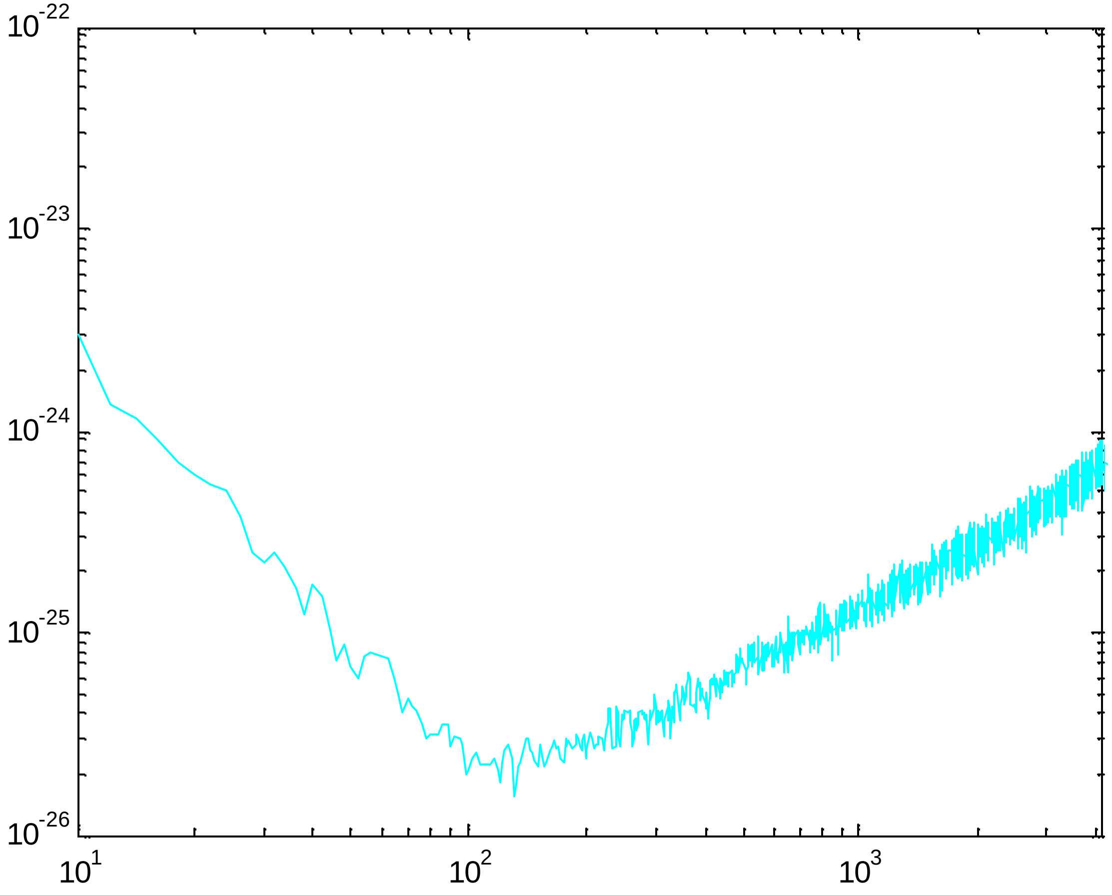


1024



4096

Colored gaussian noise (simData)



Monte Carlo of detector "events"

- Can generate, in ROOT, "events" from multiple IFOS, like:
  - » Locked IFO segments (***segment***), from ad hoc PDFs
  - » Noise events from ***sngl_burst*** triggers, random times at specified rates, ucorrelated between IFO's, random h_amp from ad hoc pdf
  - » GW signal event ***sngl_burst*** triggers, correlated between IFO's with "proper" time delay
  - » "Veto" events, random times at specified rates, ucorrelated between IFO's, random durations from ad hoc pdf (what DB table?)
- Search for coincidences, fill ***multi-burst*** triggers

LIGO

MetaDB tables currently defined

- Astrophysics events and GDS Triggers

The event tables hold results of astrophysics searches of defined sources such as inspiral, bursts, ringdowns, directedperiodic, and unmodeled-sources. The filters applied for the searches are stored in the filters tables.

- Filters

filter	*ilwd example*	*xml example*
filter_params	*ilwd example*	*xml example*

- GDS Trigger

gds_trigger	*ilwd example*	*xml example*

- Single-Inferometer Astrophysics Events

sngl_inspiral	*ilwd example*	*xml example*
sngl_burst	*ilwd example*	*xml example*
sngl_ringdown	*ilwd example*	*xml example*
sngl_unmodeled	*ilwd example*	*xml example*
sngl_datasource	*ilwd example*	*xml example*
sngl_transdata	*ilwd example*	*xml example*
sngl_mime	*ilwd example*	*xml example*

- Multi-Inferometer Astrophysics Events

multi_inspiral	*ilwd example*	*xml example*
multi_burst	*ilwd example*	*xml example*

- Data Runs

Segment DB table schema

```
CREATE TABLE segment
(
-- A "segment" is a time interval which is meaningful for some reason.  For
-- example, it may indicate a period during which an interferometer is locked.

-- Database which created this entry
      creator_db          INTEGER NOT NULL WITH DEFAULT 1,

-- Unique process ID of the process which defined this segment
      process_id          CHAR(13) FOR BIT DATA NOT NULL,

-- Segment group (e.g. 'H2-locked') and version to which this segment belongs
      segment_group       VARCHAR(64) NOT NULL,
      version             INTEGER NOT NULL,

-- INFORMATION ABOUT THIS SEGMENT
-- Segment start and end times, in GPS seconds and nanoseconds.
      start_time          INTEGER NOT NULL,
      start_time_ns       INTEGER NOT NULL,
      end_time            INTEGER NOT NULL,
      end_time_ns         INTEGER NOT NULL,
```

Sngl_burst DB table schema

```
CREATE TABLE sngl_burst
(
-- Event table for single-interferometer burst-event search.

-- Database which created this entry
      creator_db          INTEGER NOT NULL WITH DEFAULT 1,

-- INFORMATION ABOUT THE PROCESS WHICH GENERATED THIS EVENT
-- Process which generated this event
      process_id          CHAR(13) FOR BIT DATA NOT NULL,
-- Filter identifier (indicates type of filter, plus parameters).  May be null
      filter_id           CHAR(13) FOR BIT DATA,
-- Interferometer
      ifo                 CHAR(2) NOT NULL,

-- TIME OF THE EVENT
-- The start time of this burst event (in GPS seconds and nanoseconds)
      start_time          INTEGER NOT NULL,
      start_time_ns       INTEGER NOT NULL,
-- The time duration of this burst event (seconds)
      duration            REAL NOT NULL,

-- PROPERTIES OF THE EVENT
-- Center of frequency band in which observation is made (Hz)
      central_freq        REAL,
-- Range of frequency observed (Hz)
      bandwidth           REAL,
-- Absolute signal amplitude (fractional strain)
      amplitude           REAL NOT NULL,
-- Signal to noise ratio
      snr                 REAL,
-- Confidence variable
      confidence          REAL,

-- Unique identifier for this event
      event_id            CHAR(13) FOR BIT DATA NOT NULL,
```

Multi_burst DB table schema

```
CREATE TABLE multi_burst
(
-- Event table for multi-interferometer burst-event search.

-- Database which created this entry
      creator_db          INTEGER NOT NULL WITH DEFAULT 1,

-- INFORMATION ABOUT THE PROCESS WHICH GENERATED THIS EVENT
-- Process which generated this event
      process_id          CHAR(13) FOR BIT DATA NOT NULL,
-- Filter identifier (indicates type of filter, plus parameters).  May be null
      filter_id           CHAR(13) FOR BIT DATA,
-- Interferometers used for this search
      ifos                CHAR(12) NOT NULL,

-- TIME OF THE EVENT
-- The start time of this burst event (in GPS seconds and nanoseconds)
      start_time          INTEGER NOT NULL,
      start_time_ns       INTEGER NOT NULL,
-- The time duration of this burst event (seconds)
      duration            REAL NOT NULL,

-- PROPERTIES OF THE EVENT
-- Center of frequency band in which observation is made (Hz)
      central_freq        REAL,
-- Range of frequency observed (Hz)
      bandwidth           REAL,
-- Absolute signal amplitude (fractional strain)
      amplitude           REAL NOT NULL,
-- Signal to noise ratio
      snr                 REAL,
-- Confidence variable
      confidence          REAL,

-- Direction of Hanford-to-Livingston ray at time of event
-- (i.e. the central axis of the cone on which the source lies)
```

Daniel's *Event* Class to represent DB events in ROOT

Event Analysis Package.

Event Analysis Package

- [Event class]
  Event class and helpers
- [Event functions]
  Event functions
- [Event conditions]
  Event functions
- [Event containers and algorithms]
  Event containers and algorithms

The event analysis package provides means to analysis events generated by the LIGO detectors. Its major components are:

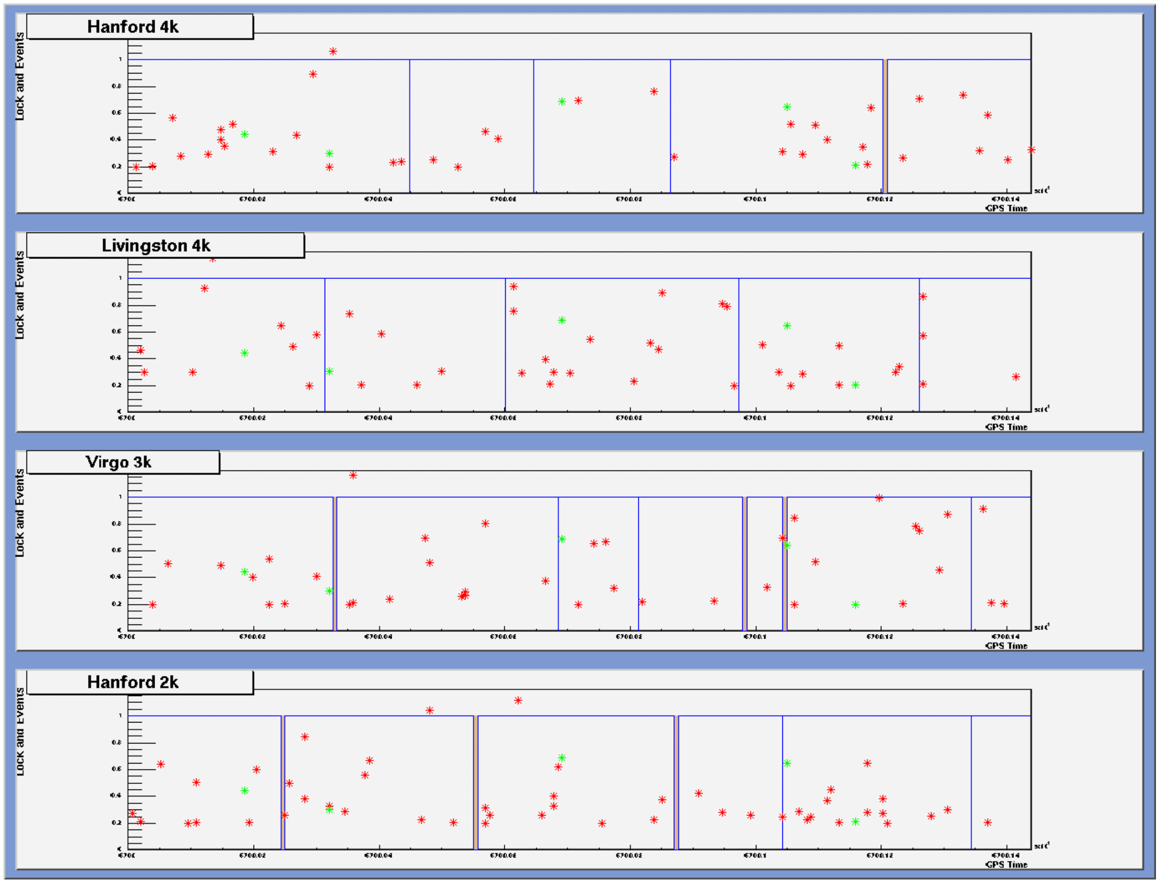
```
Event - A set of classes defining a general purpose event
Containers - A set of containers to store events
Functions - Function objects acting on events
Conditions - Conditions acting on events
Algorithms - A set of algorithms to select and histogram events
```

An event is an object consisting of a time, a type, an interferometer set and an arbitrary number of columns.

The layout of the class library is as follows

```
Type - Describes an event type
IfoSet - Describes the origin of an event
Event - The basic event object (inherits from Type and IfoSet)
EventPtr - A smart pointer for storing events
```

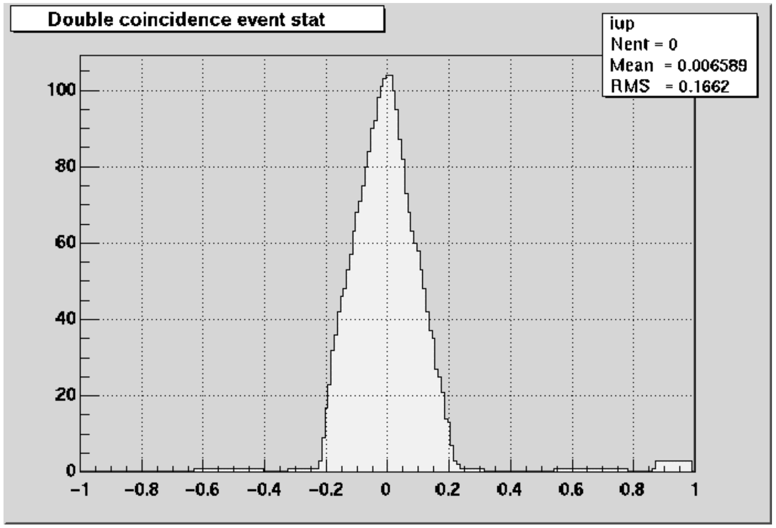
Example with 4 IFOs (not yet with ***Event*** class)



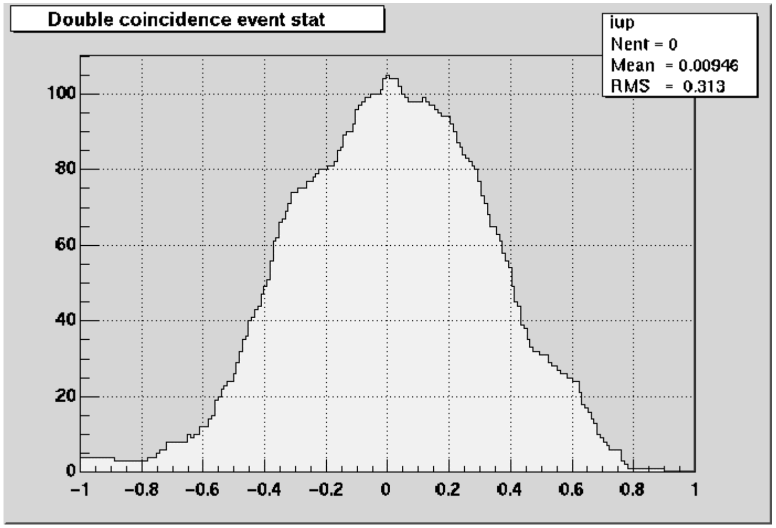
- 4 IFOs (can do bars, SNEWS, etc)
- In this example, 5 hours of data
- Locked segments are shown; brief periods of loss of lock.
- fake randoms are red; correlated GW bursts are green
- Vetoed stretches not displayed here; but available
- This is all still in ROOT; need to write ilwd, deposit into metaDB, read back into ROOT from DB, do coincidence analysys.

Delayed 2-fold coincidence analysis

L4K / H4K



K4K / VIRGO



In these examples, real event rate was very high (10/hr !), fake rate “realistic” (100/hr)

Proposed frames for MDC

WHITE NOISE
- one second of white noise sampled at 16384, stored as floating point with mean 0 and width 1, in a single frame file with one channel, channel H2:LSC-AS_Q
- same as above, 64 seconds (2^20 samples)
- same as above, 8.53 minutes (2^23 samples)

COLORED NOISE
- the same with COLORED noise

E2 NOISE
- 1 second of E2 H2:LSC-AS_Q data
- 64 seconds of E2 H2:LSC-AS_Q data

COLORED/E2/E5 NOISE with signal>RF:
- 64 seconds of white noise sampled at 16384, as above, on which is added a ZM waveform every second on the half-second filtered through the E2 transfer function and with a h_peak that is roughly X (3?) times the min noise sigma.
- same, with 100 msec ring-downs (f0 ranging from 100 to 300 Hz).

The big question

- How best to characterize waveforms and our response to them in an astrophysically meaningful way? h_{rms}, Δt, $[f_0, f_0+\Delta f]$
- Some “inner product” of filter to waveform?
- …

Many little questions

- How long a data stretch should we analyze in one LDAS job? (Inspiral people use 2^{23} samples = 8.533 minutes…)
- How much (should we) decimate from 16384 Hz? Inspiral people decimate down to 1024 or 2048; there is little inspiral power above 1 kHz. Not so for millisecond bursts! (Bar people look for delta function – a single ADC count).
- How much overlap should we include?
- What's the best way to insert fake signals? Randomly in time? With/without antenna pattern? How to systematically explore parameter space?
- Where/when do we fully whiten the (somewhat whitened) data?
- At what stage do we apply gross vetos (IFO in lock), finer vetos (coincidence with PEM event), etc?
- How to package TF curve with data in frames?
- …