LIGO

Vetoes used in the
Burst and Inspiral Analyses
Part 1

Stefan Ballmer

Laura Cadonati

Erik Katsavounidis

Peter Shawhan

LIGO Lab

LIGO-G020355-00-Z

Vetoes Investigators

Saulson, Zweizig, Christensen, Schofield, Shawhan, Ito, Sigg, Rahkola, Cadonati, Ballmer, Gonzalez, Sylvestre, Shoemaker, Weinstein, Vijay, Klimenko, Katsavounidis, Camp (ca. March 2002)

- Identify software needs for doing the vetoes investigations
- Set the goals for the veto effort
- Deliver the vetoes for the E7 analysis

Overview

- Burst and Inspiral DSOs during E7 data taking were generating GW triggers at rates [O(1Hz)]
- DSO-based criteria alone could further eliminate triggers, but still not enough
- E7 data taking was 'bursty'
- Goals for a veto strategy
  - Define auxiliary channels that show 'burstiness' that correlates to transient behavior in the AS_Q
  - Use these channels to reduce (veto) the GW candidates
  - Keep as much of the detector's livetime as possible
  - Optimize the choice of veto channels, glitch finding method, thresholds of the veto 'significance'

Search for good veto channels

- **From hand-scanning we knew:**
  - PEM channels useless for E7 data (event rate too high)
  - Typical glitches: L1: PSL glitch, H2: LSC servo glitch

- **Selected 8 IFO channels for detailed investigation:**

H2:LSC-MICH_CTRL	L1: LSC-MICH_CTRL
H2:LSC-CARM_CTRL	L1:LSC-CARM_CTRL
H2:LSC-REFL_Q	L1: LSC-REFL_Q
H2:LSC-POB_Q	L1 :PSL-FSS_RCTRANSPD_F (mislabeled)

- **Chose 2 different filters: 30Hz and 100Hz high pass**

- **Ran glitch finder (absGlitch) over ~3hrs of data (playground data set)**

absGlitch (R.Rahkola, M.Ito)

- **absGlitch Monitor:**
  - Applies IIR filter
  - Finds (absolute) threshold crossings
  - Last crossing = signal below threshold for next 0.25sec (tunable)
 - Trigger reported from 1st to last threshold crossing
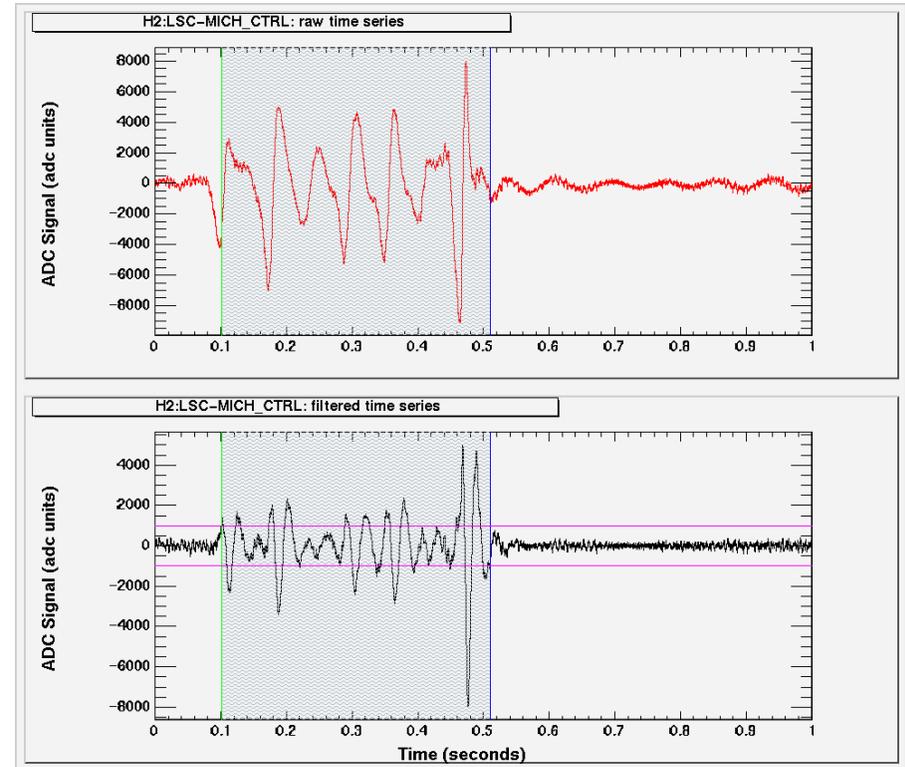
Other Monitors

- Gide (J.Sylvestre)
  - Matched filtering for typical glitch
- PSLmon (J.Zweizig)
  - Uses floating threshold
- ➔ Both give results similar to absGlitch
- Inspiral template triggers (ran on veto channel)
  - Used by Inspiral Group (better than absGlitch)
  - Burst Search:
    - ~same Efficiency (for same dead time)
    - but seem to be orthogonal to absGlitch triggers

Defining Veto's Time Window

- absGlitch reports trigger duration

➔ Veto event if it overlaps with trigger
 - Used in E7 analysis
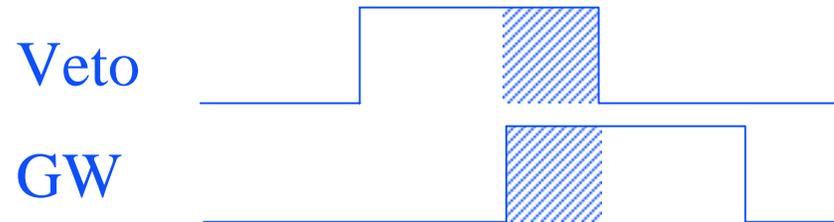
- Other definitions possible:
  - E.g. fixed length window 'triggered' by the veto
  - Fixed length window 'centered' at the veto (IUL)
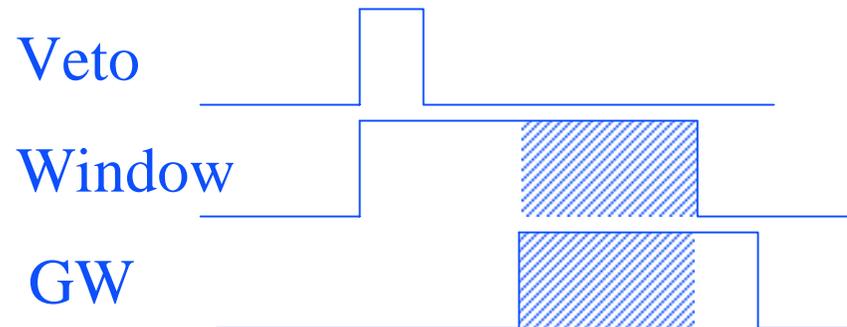
Veto Efficiency vs Dead time

- Apply vetoes to TFCLUSTER event candidates
- → 30Hz HP filter better than 100Hz (more tuning possible)
- L1:
  - → Best channel:
    - – L1 :PSL-FSS_RCTRANSPD_F
  - → Best threshold:
    - – L1-PSL30: 12 ADC counts (Burst)
      - → minimizes residual rate
    - – L1-PSL30: 10 ADC counts (Inspiral)

Note: now only 256Hz sampling rate

Legend:
- L1-PSL30
- L1-CARM30
- L1-MICH30
- L1-REFL30
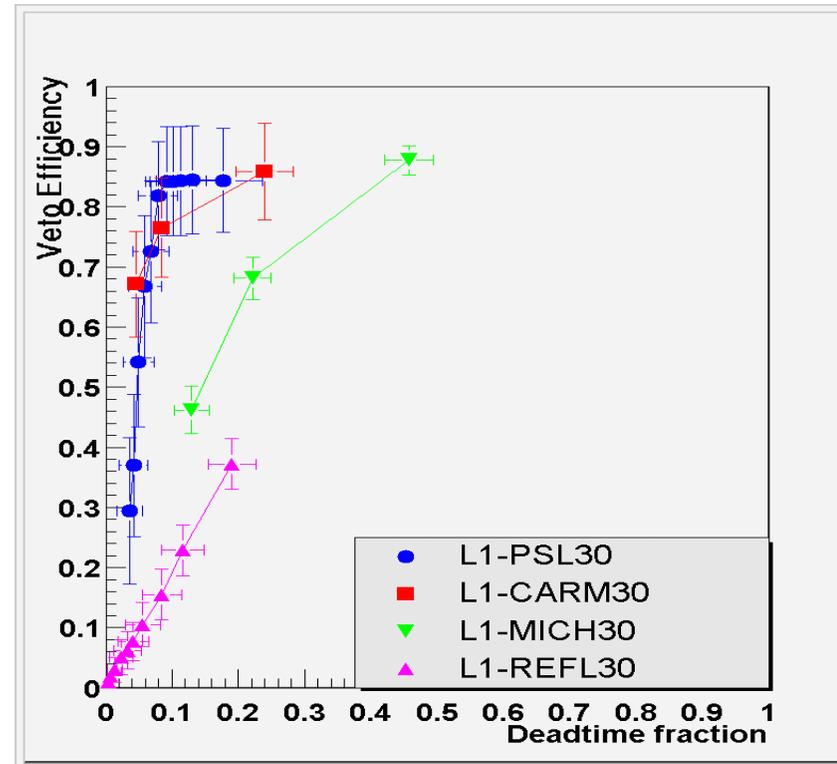
- Veto Eff. = Fraction of vetoed TFC events
- Dead time = Sum of veto durations
- Plotted for different absGlitch thresholds

Vetoing by chance: Veto Efficiency after Time Shift

Plot Veto Efficiency vs. artificial time shift:

● Peak:
  ▪ True Veto Efficiency
  ▪ Max for Threshold <= 12

● Wings:
  ▪ coincidence by chance
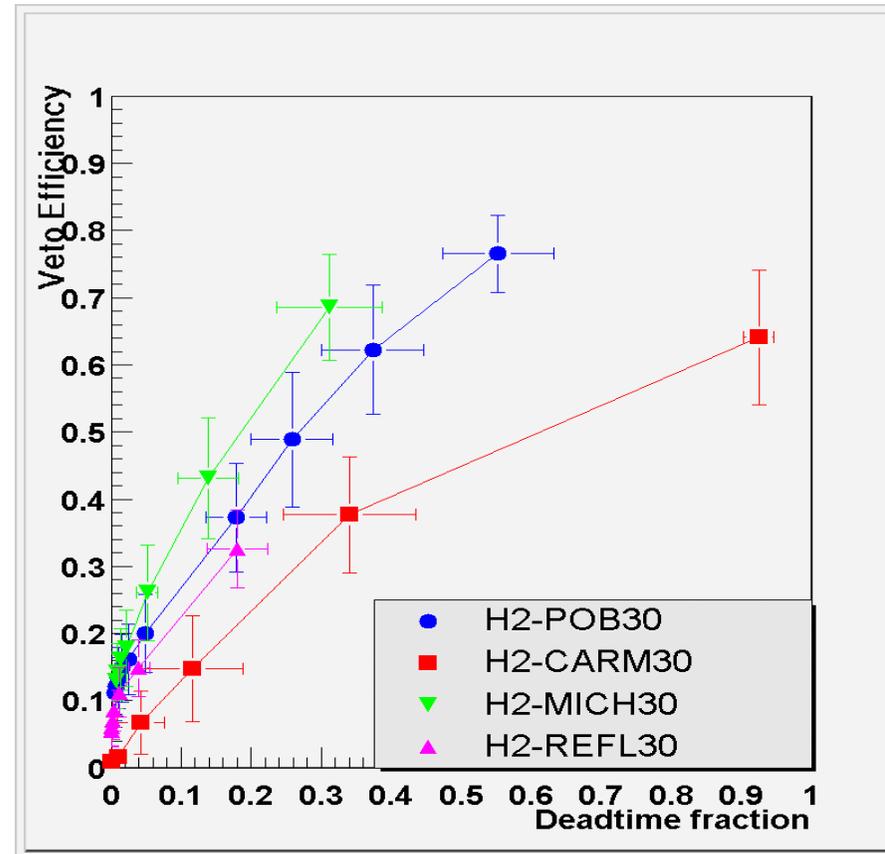 ▪ Lowering threshold increases wings

➔ Threshold = 12 optimal

Veto Lag Plot at L1 - PSL 30HzHP

absGlitch Threshold: 8, 10, 12, 16, 20, 25, 30
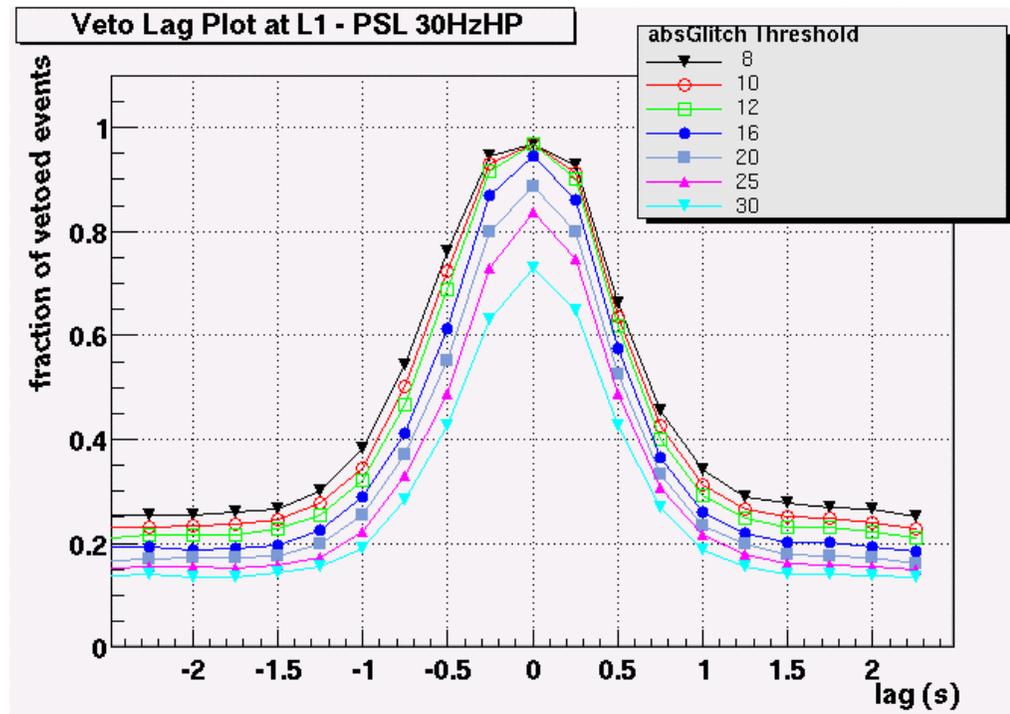
Veto for H2

- ➔ Best channel:
  - - H2:LSC-MICH_CTRL
- ● H2 Veto not as good as L1
- ● But picks out whoppers
- ➔ Chosen threshold (Burst):
  - ▪ H2-MICH30: 2000 ADC counts
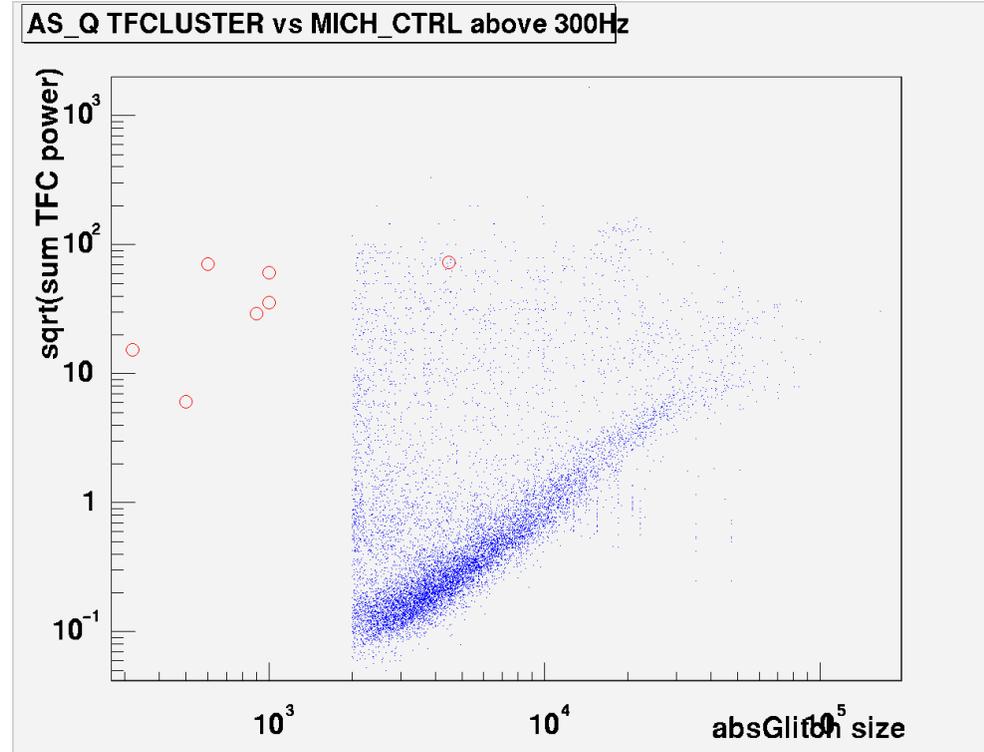- ➔ Inspiral Group uses matched templates

BUT:

- ● Coupled to AS_Q
  - ● Vetoing GW events?

GW in MICH_CTRL ?

- Lm excitations DO show up in MICH_CTRL, but

- AS_Q/MICH_CTRL is ~100x bigger than for the frequent servo glitches

→ Only reject events with small enough AS_Q/MICH_CTRL ratio

AS_Q TFCLUSTER vs MICH_CTRL above 300Hz

- Plotted for E7 data
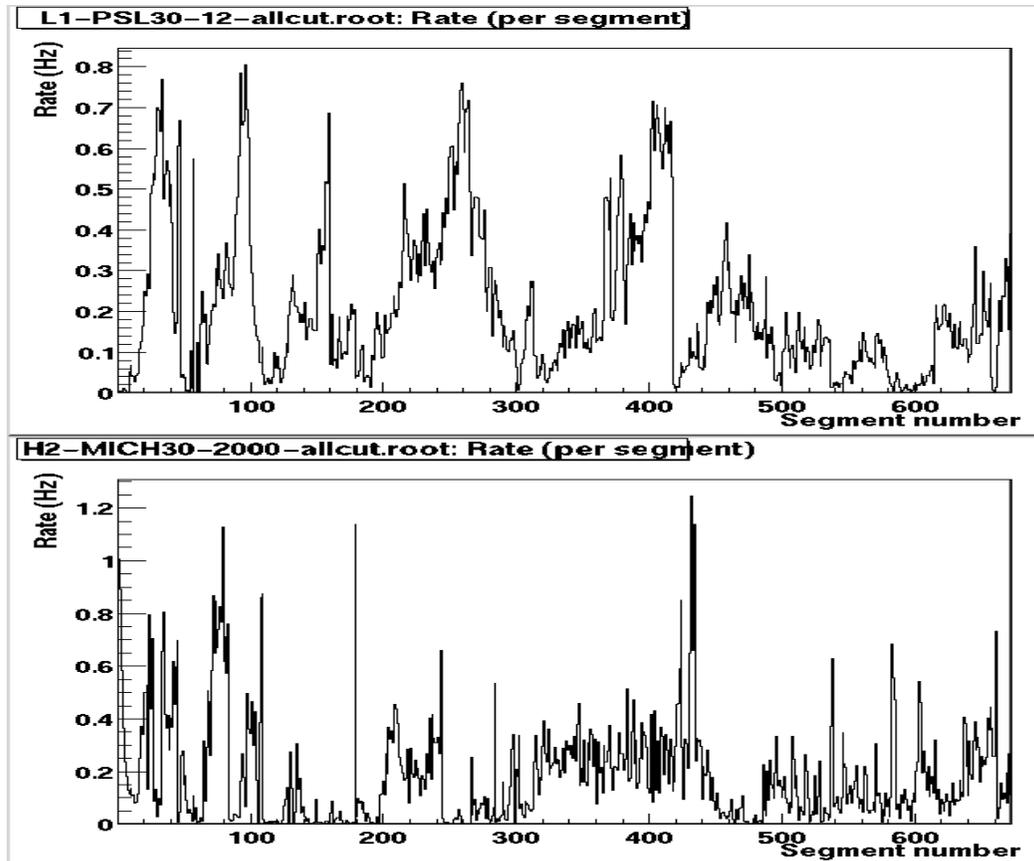
Full E7 absGlitch Triggers
Trigger Rate per Locked Segment

LIGO

L1:

- relatively smooth variations (lock independent)

H2:

- ~20 very noisy segments

- Big segment to segment variations

L1-PSL30-12-allcut.root: Rate (per segment)

H2-MICH30-2000-allcut.root: Rate (per segment)

Full E7 absGlitch Triggers Dead Time per Locked Segment

- absGlitch triggers can be quite long

➔ More accurate measure than rate:
  - Dead Time fraction

L1-PSL30-12-allcut.root: Deadtime fraction (per segment)

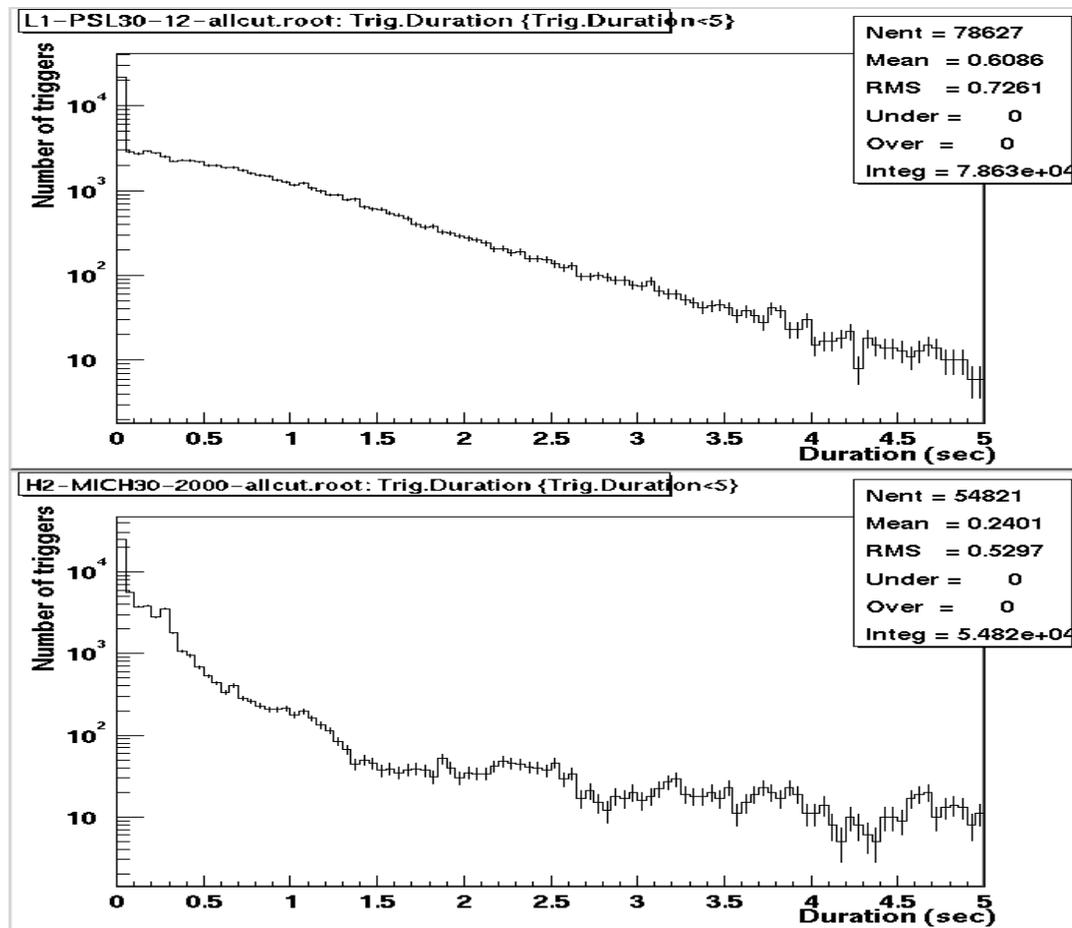H2-MICH30-2000-allcut.root: Deadtime fraction (per segment)
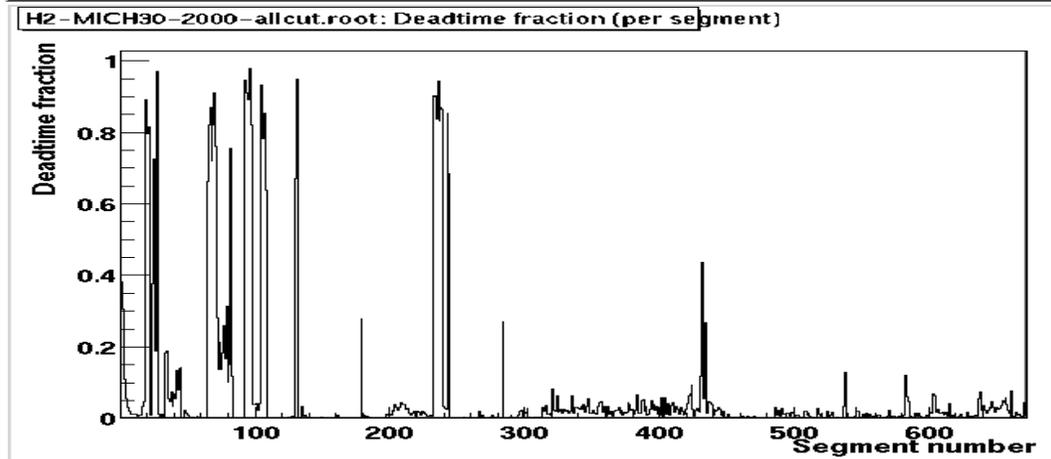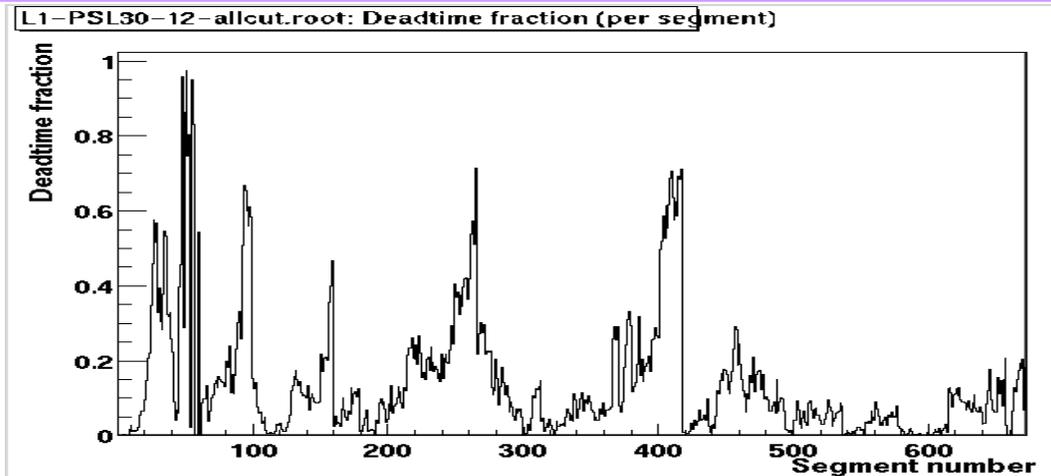
Full E7 absGlitch Triggers
Time between triggers

- Triggers cluster on time scales <~5sec (~trigger duration)

- MICH_CTRL shows ~0.8sec periodicity (also seen in AS_Q)
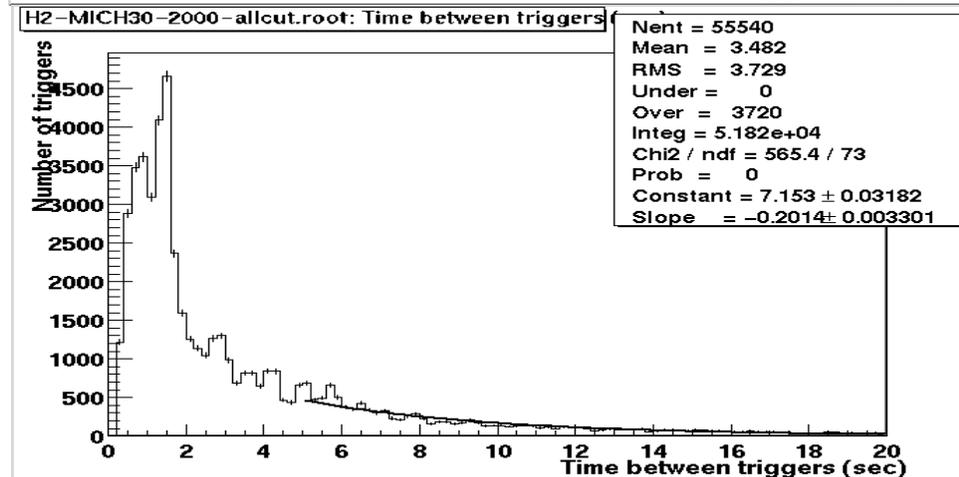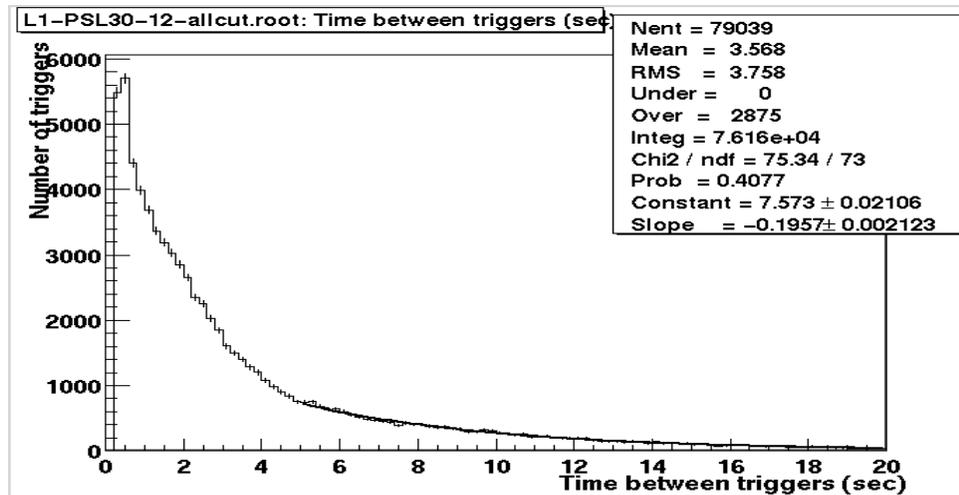
Full E7 absGlitch Triggers Trigger Durations

- **L1: smooth distribution**

- **H2: more bumpy (due to oscillations?)**

- **Excess triggers with duration <~1/30Hz**

- ➔ **single oscillation crosses threshold**

L1-PSL30-12-allcut.root: Trig.Duration {Trig.Duration<5}

Nent = 78627
Mean = 0.6086
RMS = 0.7261
Under = 0
Over = 0
Integ = 7.863e+04

H2-MICH30-2000-allcut.root: Trig.Duration {Trig.Duration<5}

Nent = 54821
Mean = 0.2401
RMS = 0.5297
Under = 0
Over = 0
Integ = 5.482e+04

Conclusions

- Best channel:
  - L1 :PSL-FSS_RCTRANSPD_F (now only 256Hz)
  - H2:LSC-MICH_CTRL
- Used absGlitch with 30Hz HP filter
  - Good choice, but more tuning possible
- Inspiral Group used matched template veto for H2

- MICH_CRTL with ratio check usable as veto channel

- http://ligo.mit.edu/ldas/research.html