SLOPE: A MATLAB Revival

Amber L. Stuver
stuver@gravity.psu.edu

Outline

- Introduction
- How Does it Work?
  - » Description
  - » Trigger Generation & Clustering
- Results
  - » Data Conditioning
  - » False rates
  - » Efficiency and false alarms
- Future Work

Introduction

- SLOPE:
  - » is a simple algorithm
  - » is not very computationally expensive
  - » was previously applied to S1 analysis
    - – issues with background rate excluded it from upper limit estimation
  - » time domain algorithm

- Motivation:

The SLOPE results provide a valuable cross check, and we intend to continue to employ and improve the SLOPE pipeline in future analyses (Sec. VII B).

- Burst S1 paper

LIGO-G040316-00-Z

Description

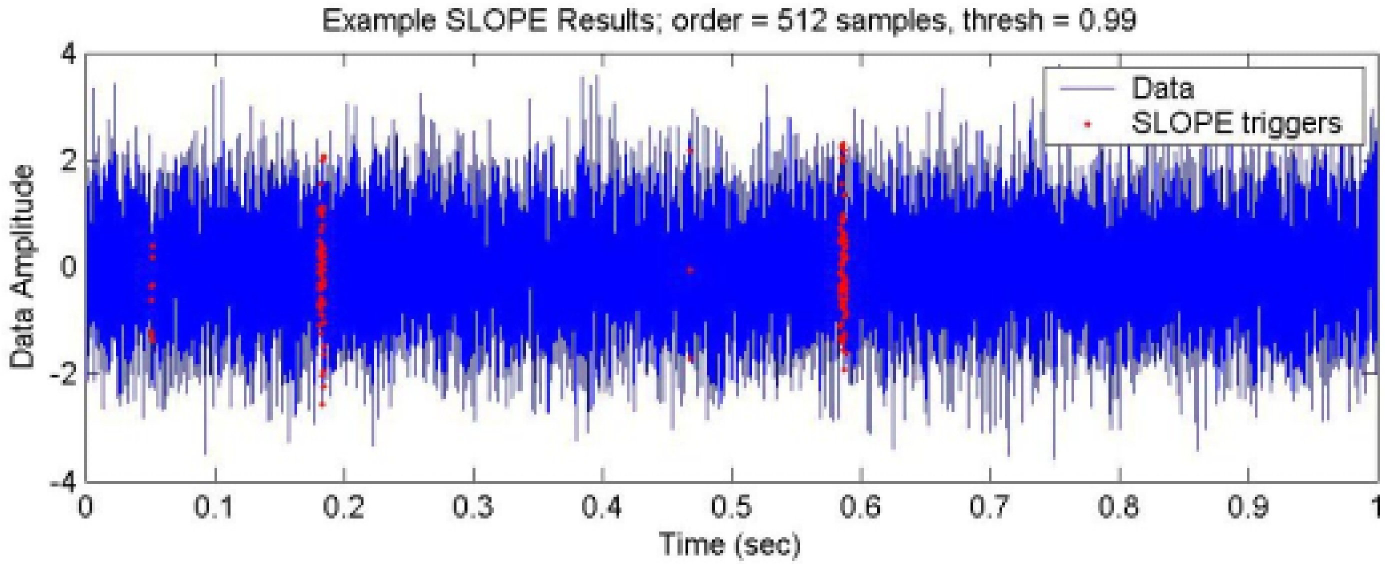
- SLOPE is a linear filter that returns the best fit slope to data
- The data is windowed according to the expected timescale of a burst signal
- This sliding window is moved through the data
- An event will typically cause a cascade of triggers
- These triggers are clustered together to form the candidate event

Trigger Generation

- SLOPE previously did not apply an adaptive threshold
- This fixed threshold was attributed to the background estimation problems in S1
- SLOPE now applies a threshold based on the idealized white noise probability that a slope that passes threshold is not accidental
- This probability threshold relates directly to the expected false rate for white noise

Trigger Clustering

- If two triggers are separated by less than the width of the window, the two triggers are considered part of the same event
- These clusters are returned as candidate events

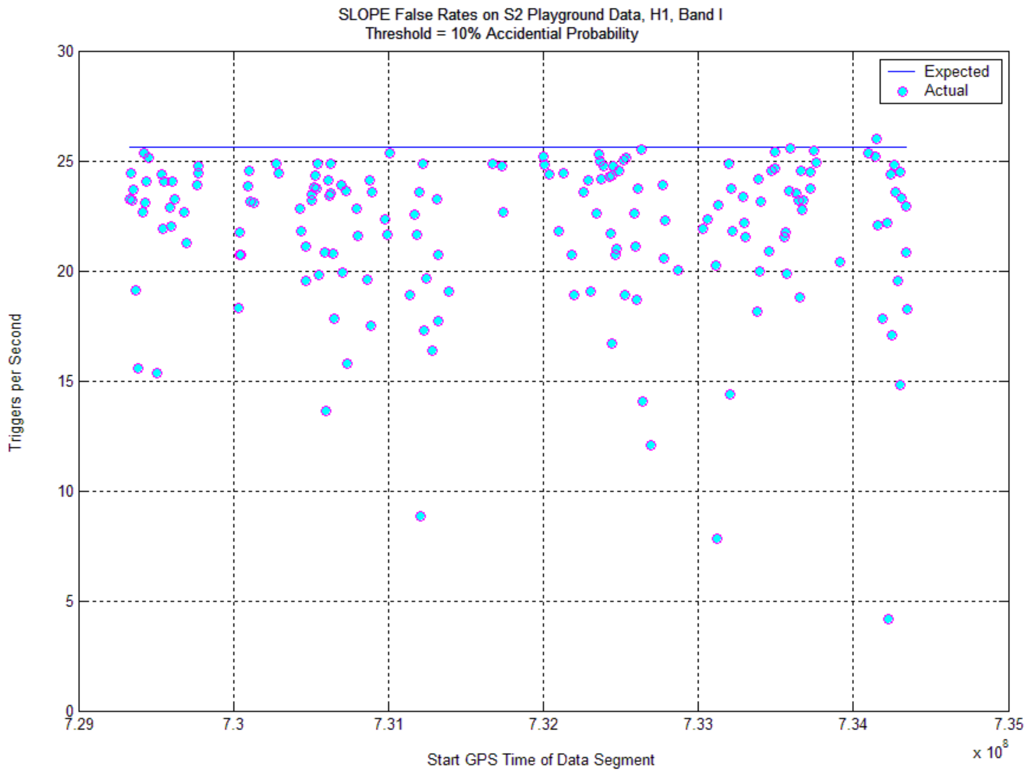


Data Conditioning

- Since SLOPE is a time domain ETG, data conditioning is necessary
  - » uses the same data conditioning as BlockNormal
- The data is
  - » down sampled and Kalman filtered to remove narrow line features
  - » base banded
  - » regressed to remove power lines and calibration lines
  - » whitened
- Results presented here are in the band between 512 to 640 Hz

False Rates

- S2 playground false rates track the ideal white noise rate
- Departures from the ideal false rate indicate non-stationarities
- False rates can be improved with applying SLOPE to stationary epochs



LIGO-G040316-00-Z

Efficiencies

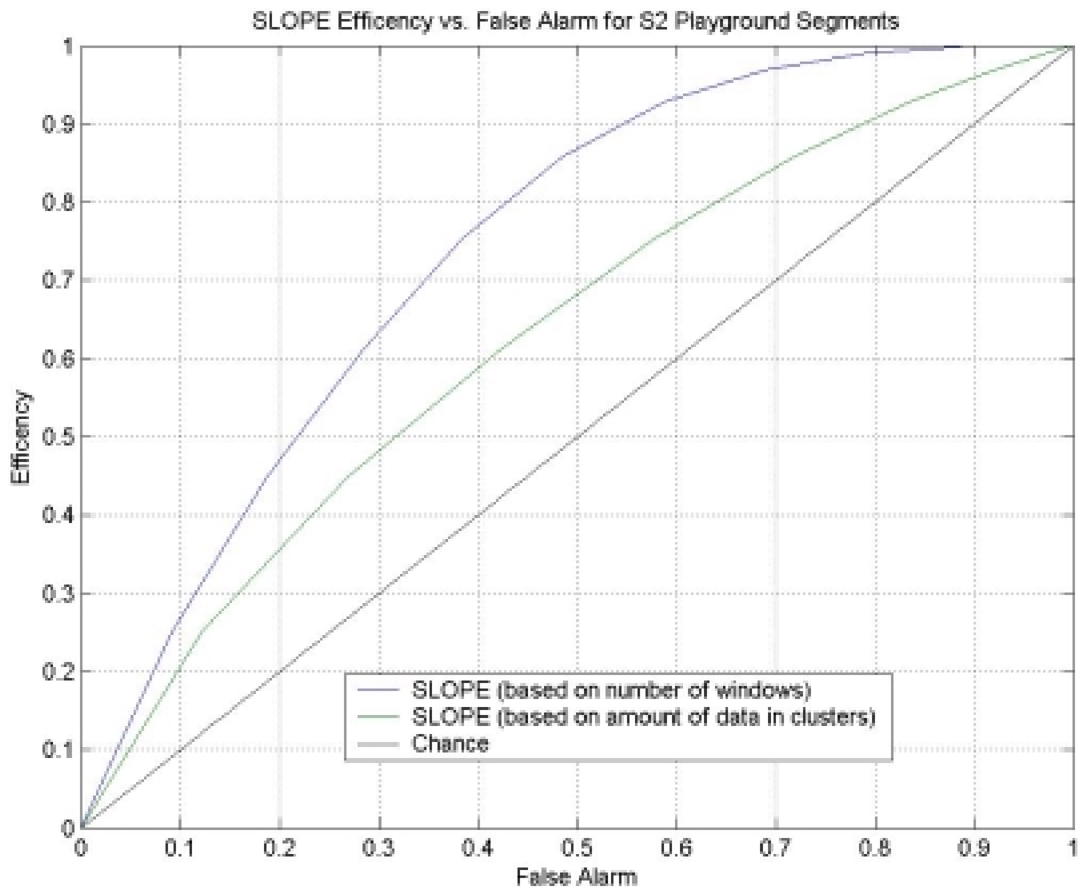
- GravEn was used to simulate sine-Gaussians at 153 Hz, Q=10
- Simulations were injected at an average rate of 1 Hz
- Criteria for detection:
  - » trigger cluster must overlap with the time of maximum amplitude of simulated signal
- Efficiency:

$$\text{efficency} = \frac{\text{number detected}}{\text{total simulations}}$$

- False Alarm:

$$\text{false rate} = \frac{\text{number of samples within clusters}}{\text{total number of samples}}$$

Efficiency vs. False Alarm



LIGO-G040316-00-Z

Future Work

- SLOPE is to be included in my thesis investigation on the efficacies of the different burst ETG's for different waveform properties
- Break data up into stationary epochs to further improve false rate and efficiency
- Investigate:
  - » effects of window size on efficiency
  - » different clustering criteria (i.e. different minimum distance between triggers)
- Suggestions welcome!