

バースト解析パイプラインの開発

端山和大、KAGRA バーストグループ

2014年7月25日

目次

| | | |
|-----|--------------------------------|---|
| 1 | データ取得部 | 2 |
| 1.1 | 解析する時刻の指定 | 2 |
| 1.2 | Frame ファイルからデータを読む | 2 |
| 1.3 | データクオリティの評価 | 2 |
| 2 | データコンディション部 | 2 |
| 2.1 | データの評価 | 2 |
| 2.2 | データのガウス性評価 | 2 |
| 2.3 | KAGRA 発: ノイズモデリング | 2 |
| 2.4 | ライントラッキング | 2 |
| 2.5 | ホワイトニング | 2 |
| 3 | Event Trigger Generation 部 | 3 |
| 3.1 | Time-Frequency Map | 3 |
| 3.2 | Clustering | 3 |
| 3.3 | Statistics for event selection | 4 |
| 4 | Upper limit or Detection | 4 |
| 4.1 | Background Analysis | 4 |
| 4.2 | Veto Analysis | 4 |
| 4.3 | Upper limit or Detection | 4 |

1 データ取得部

1.1 解析する時刻の指定

- (実装) UT,MDJ などと GPS の相互変換
- (実装) GPS 時刻から対応するデータが入っている Frame ファイルを抽出。更新ファイルの表示。

1.2 Frame ファイルからデータを読む

- (実装) データのロード
- (実装) チャンネル名、サンプリングレートの取得

1.3 データクオリティの評価

Detector characterization の結果を利用する。主に Data quality flag によるデータ選択。

- **KAGRA 発:** 最適ネットワーク選択

2 データコンディション部

2.1 データの評価

Detector characterization の結果を利用する。

2.2 データのガウス性評価

(実装) Rayleigh 分布チェック。

2.3 **KAGRA 発:** ノイズモデリング

(実装) Student-t によるノイズモデリング。

2.4 ライントラッキング

LIGO メンバ (Soumya Mohanty,...) との共同研究。

2.5 ホワイトニング

2.5.1 Power spectrum density

(実装) Median-average 法、Welch 法

2.5.2 Frequency-domain whitening filter

(実装) interpolation (Linear, Cubic spline)

2.5.3 Time-domain whitening filter

Linear prediction error filter の実装。

- (実装) FIR, IIR filter (C, Haskell でラッパー)。現在 Butterworth, Chebyshev1 型の Low/High pass filter を実装。
- (実装) Levinson-Durbin recursive method (C と Haskell で実装。Numerical Recipes を利用)

3 Event Trigger Generation 部

3.1 Time-Frequency Map

- Wavelet packet (coherent waveburst で採用。計算効率、実現できる解像度を考えるとバースト解析においてベストに近い。)
- **KAGRA 発:** Q transform on nonstationary Gabor Frame。Q transform は LIGO で開発された Omega pipeline や Q pipeline で採用されている。採用された当時は Q transform には精度の良い逆変換が構成できていなかった。そのため時間周波数空間に行ったら帰ってこれないという強い制限があった。この制限により、Q transform 後に時間周波数空間上で行うフィルタ処理の結果を用いた解析や位相を含めた解析ができなかった。2011 年に調和解析の分野で進展があり、nonstationary Gabor frame を導入することで逆変換が構成できるようになった。これを実装することを検討している。これにはいくつかのアドバンテージがある: LIGO への直接的な貢献をする。Wavelet 系では弱かった周波数解像度を効率よく上げられる。SFTF 系では time-frequency atom が固定されていたのが、multi-resolution atom にできる。

3.2 Clustering

3.2.1 Signal dependent modeling

今まで行われてきた方法。信号依存が強いので基本的にアドホックにならざるを得ない。最適化の方法があるの？

3.2.2 **KAGRA 発**: Sparse synthesis modeling

やられていない方法。信号を時間周波数空間で、基底がなるべく少ない形で表現したい。そのために、基底数にペナルティを課すような regulator を Tikhonov regularization か、Lasso を用いて実装したらどうか？

3.3 Statistics for event selection

3.4 データベース

そろそろスタートしないといけない。

3.4.1 Detection Statistics

4 Upper limit or Detection

4.1 Background Analysis

4.2 Veto Analysis

4.3 Upper limit or Detection