

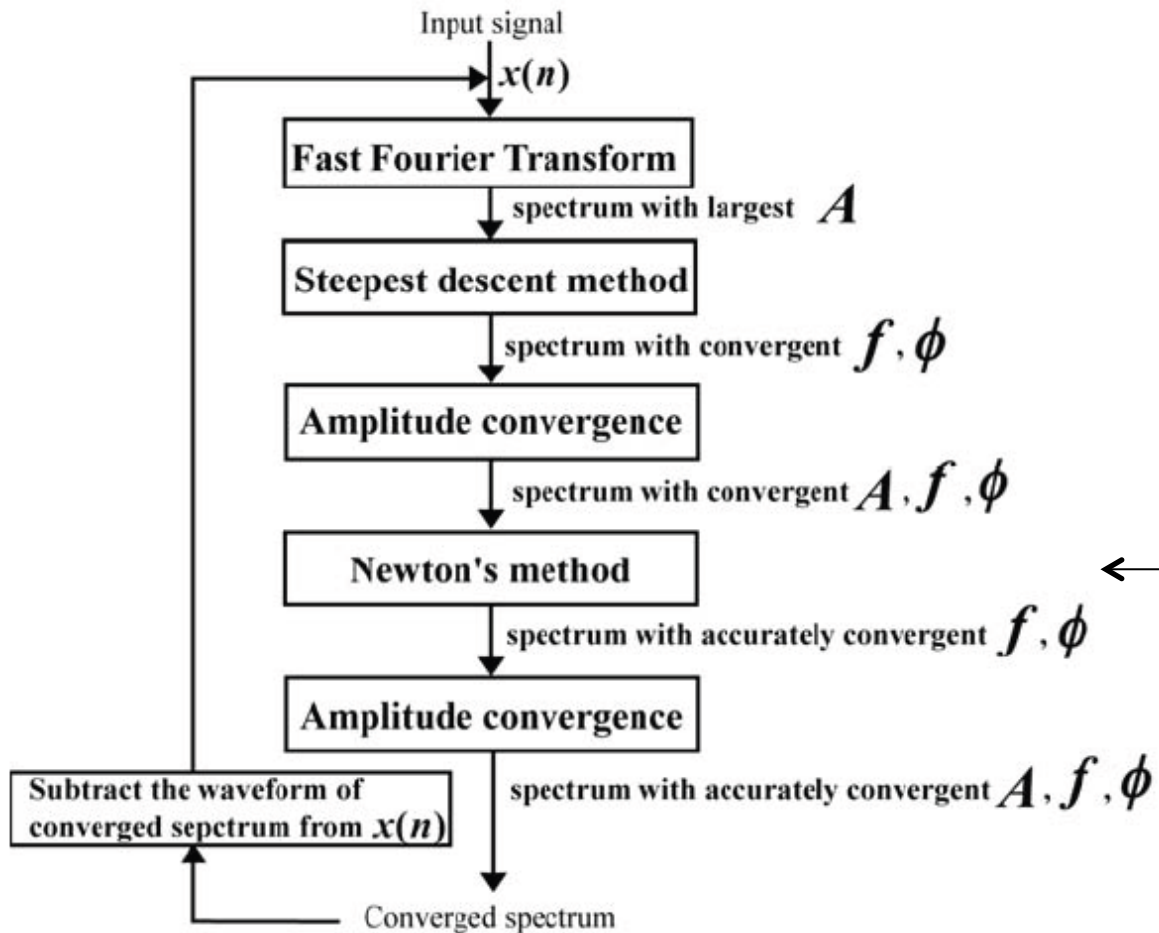
Line characterization

13 Jan. 2015

K. Ueno (Osaka Univ.)

NHA のアルゴリズム

Yoshizawa+ '11



自分のコードの中で
この辺りの計算を改良

計算コスト (新学術マシン)

$$T_{data} \cong T_{frame} \times \frac{N_{data}}{N_{shift}}$$

改良前

$$T_{frame} \cong 0.35s \times \frac{N_{frame}}{128} \times \frac{N_{spec}}{10}$$



改良後

$$T_{frame} \cong 6ms \times \frac{N_{spec}}{10} @ N_{frame} = 128$$

T_{data} : time to process given data

N_{data} : data length

N_{shift} : shift length (frame interval)

T_{frame} : time to process one frame

N_{frame} : frame length

N_{spec} : #spectra to be extracted

リアルタイム ラインキャラクターリゼーション

- 400 Hz 前後のラインを解析する場合,
 - $f_s = 1024$ Hz
 - $N_{\text{shift}} = 16$
- この条件下で 128 秒間のデータは
 - 1 CPU でおよそ 50 秒
- リアルタイム解析可能.

性能評価

入力信号

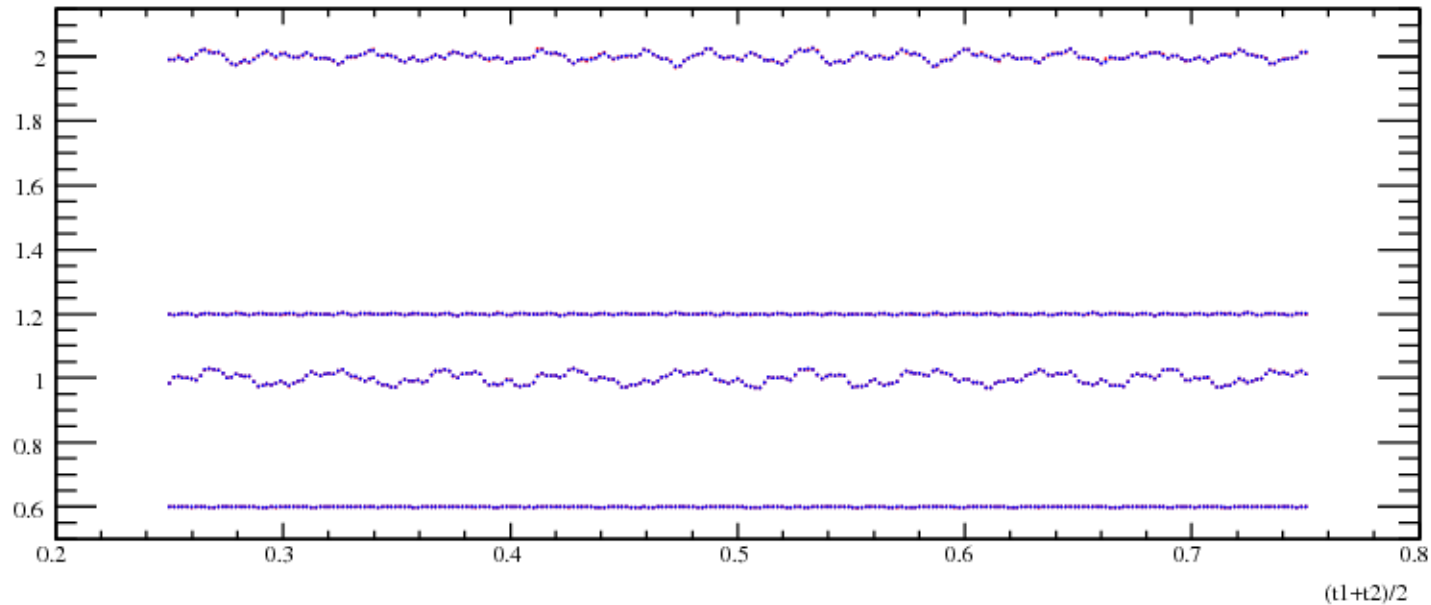
- 以下の4つの異なる正弦(余弦)関数からなる時系列を用意
- サンプリングレートは 512 Hz、時間は [0, 1] 秒を仮定

	振幅	周波数 [Hz]	初期位相 [rad]
A	2.0	66.7	-0.15
B	1.2	109.0	0.46
C	1.0	40.4	2.40
D	0.6	21.3	0.10

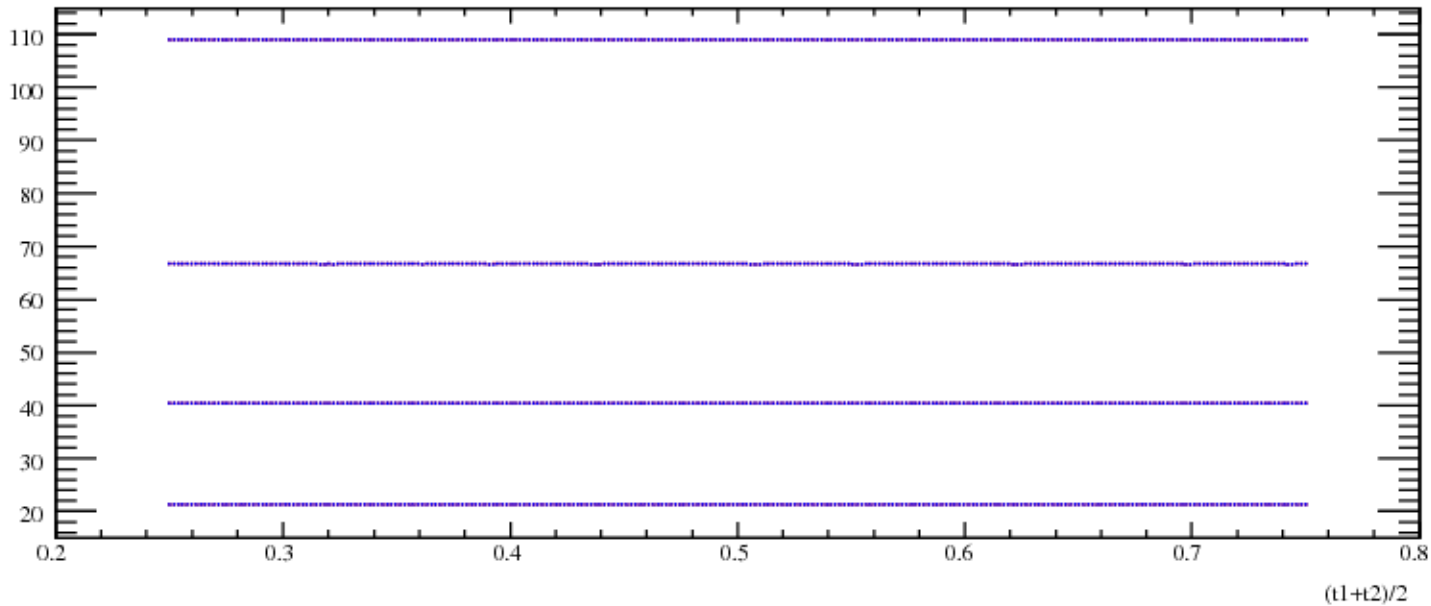
- この時系列データを富山大(中野氏)でも NHA 解析してもらい、双方の解析結果を比較

フレーム長256

振幅



周波数



赤: 富山大
青: 上野

CPU プロセス時間

- 富山大(中野氏)

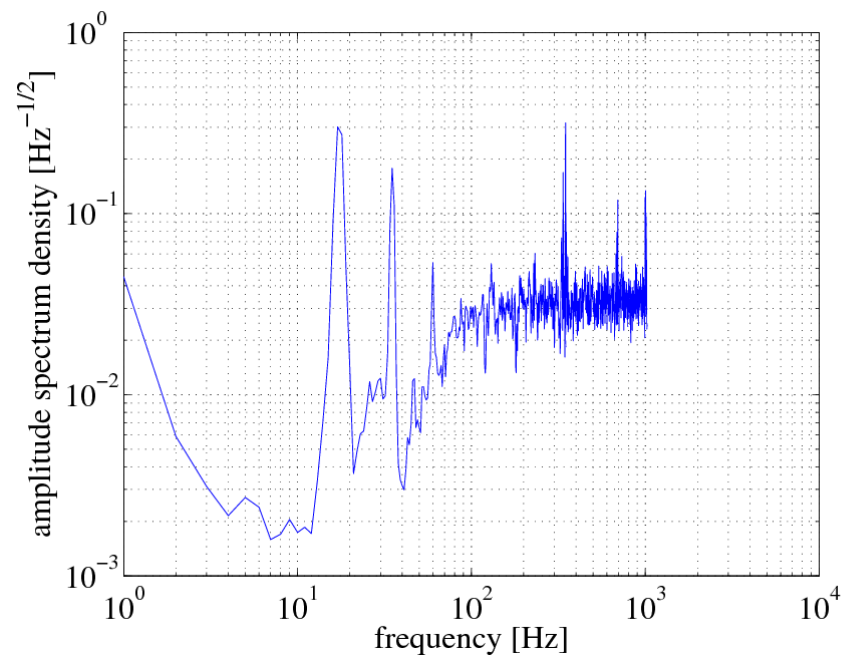
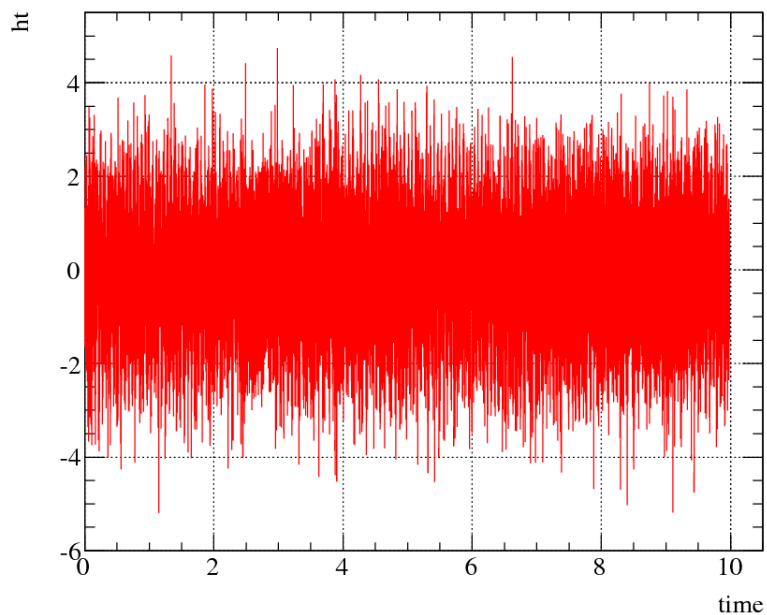
フレーム長	プロセス時間 [秒]
64	1.082
128	1.098
256	1.136

- 上野の方は、精密な時間計測は行っていない
 - 3パターンすべてが1秒程度で終了したので、ほぼ同等

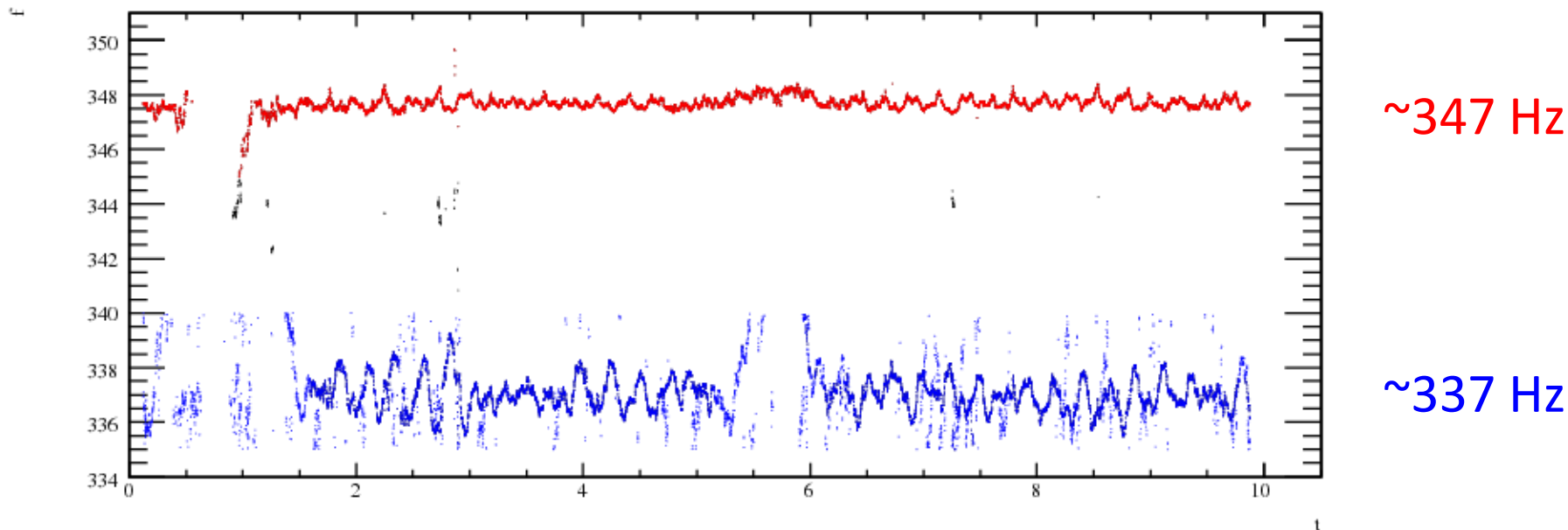
モジュレーションスタディ

LIGO S5 (H1) データ

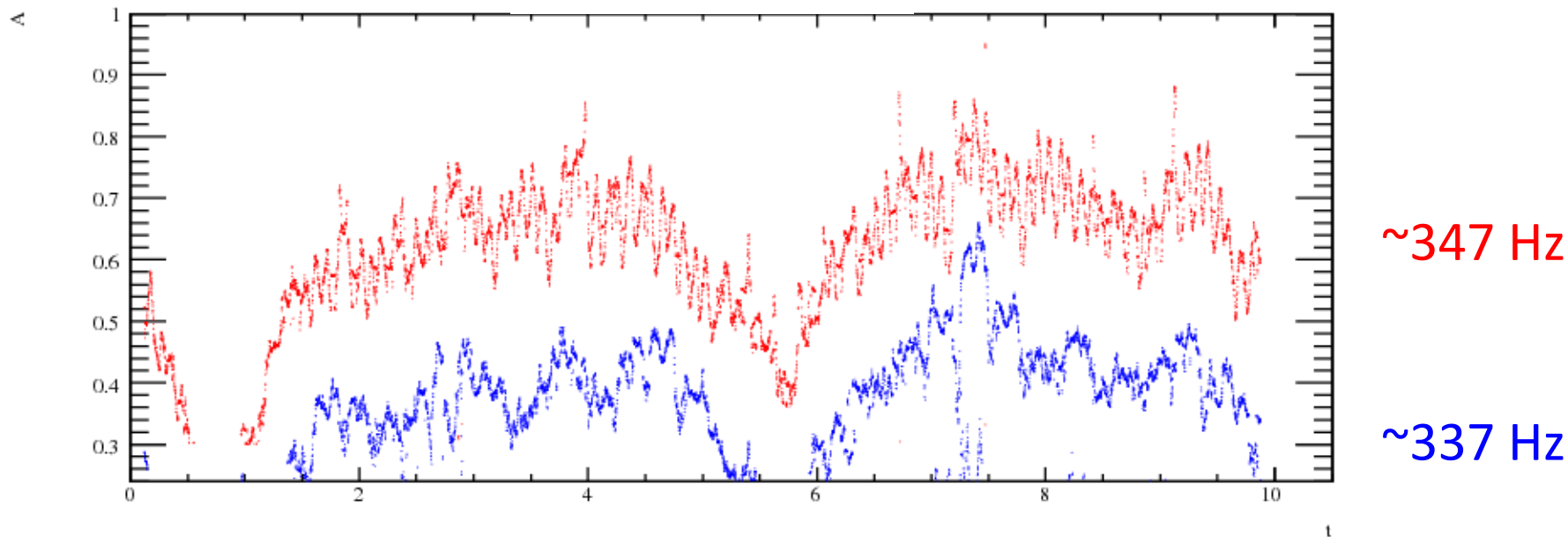
T=10 秒, $f_s=2048$ Hz



周波数



振幅



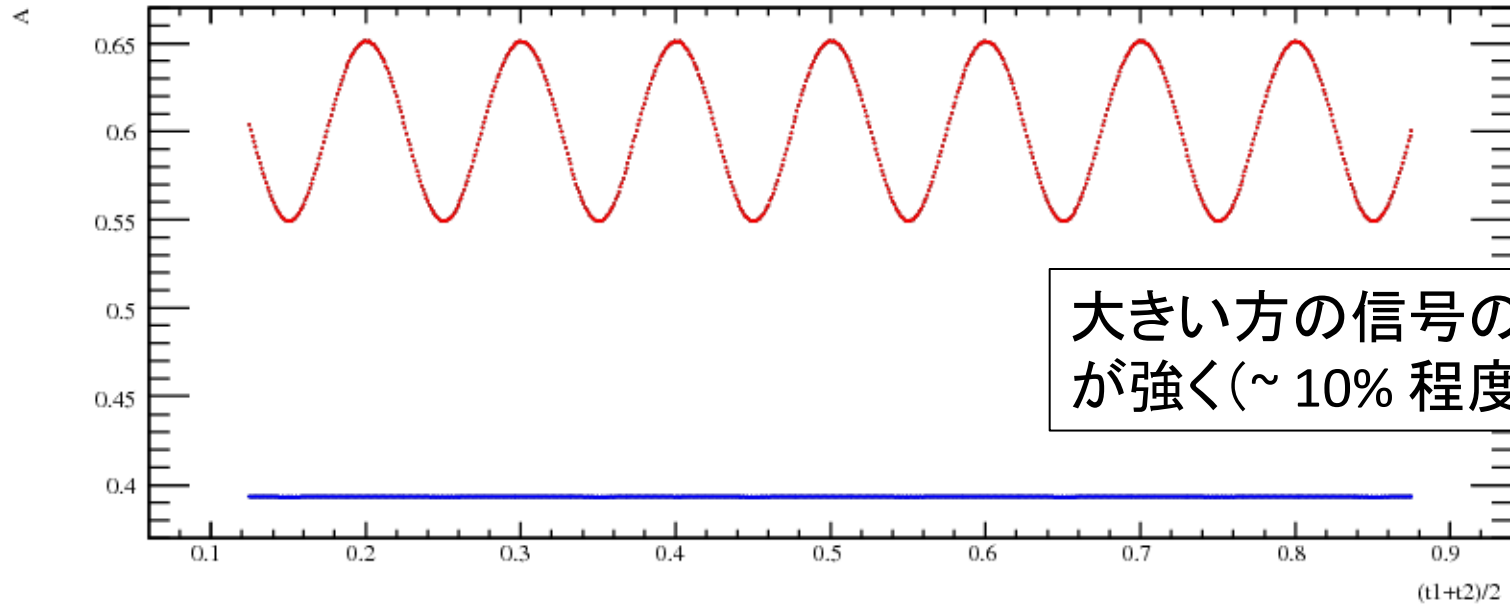
入力信号(2)

- 以下の2つの近い周波数をもつ正弦(余弦)関数からなる信号を用意
- サンプリングレートは 1024 Hz、時間は [0, 1] 秒を仮定

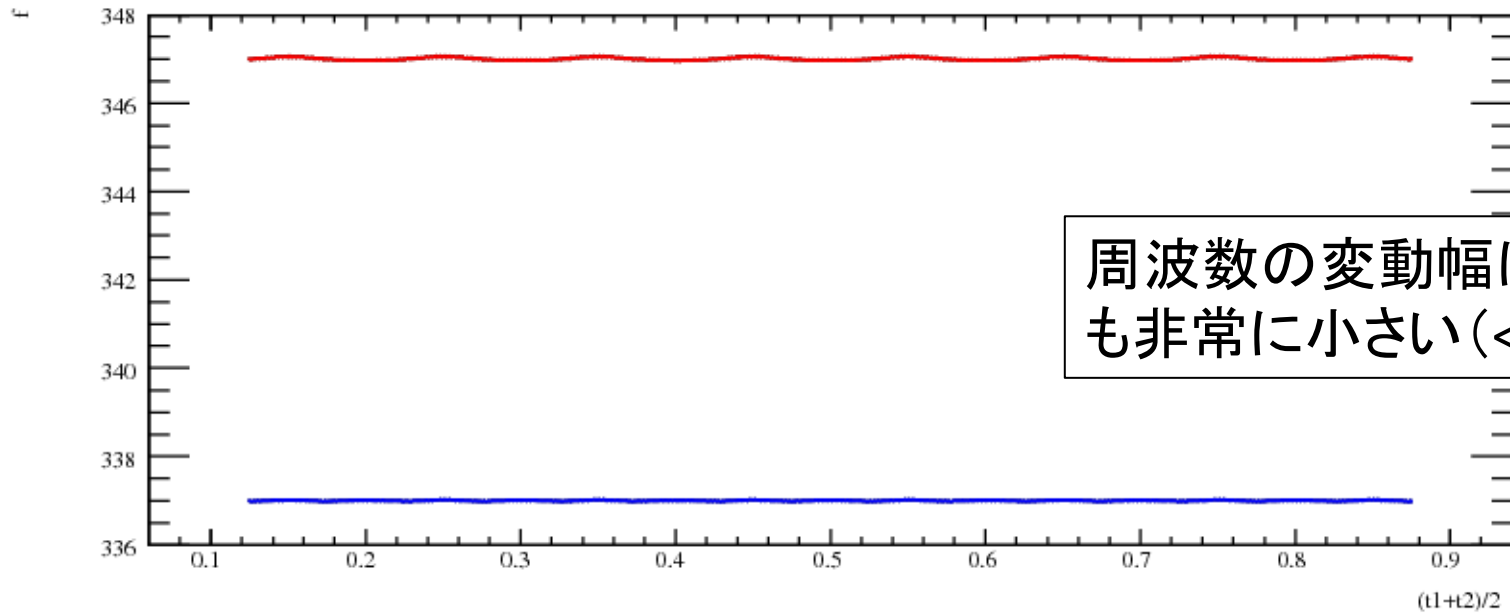
	振幅	周波数 [Hz]	初期位相 [rad]
A	0.6	347	0.4
B	0.4	337	0.4

- これにより周波数の近い信号どうしがモジュレーションを引き起こす可能性をチェックする

振幅



周波数



バックアップ

NHA のアルゴリズム

- 1) FFT the given time series $x(n)$ and find the frequency which gives the largest amplitude.
- 2) You somehow minimize the following cost function about A, f , and ϕ ,

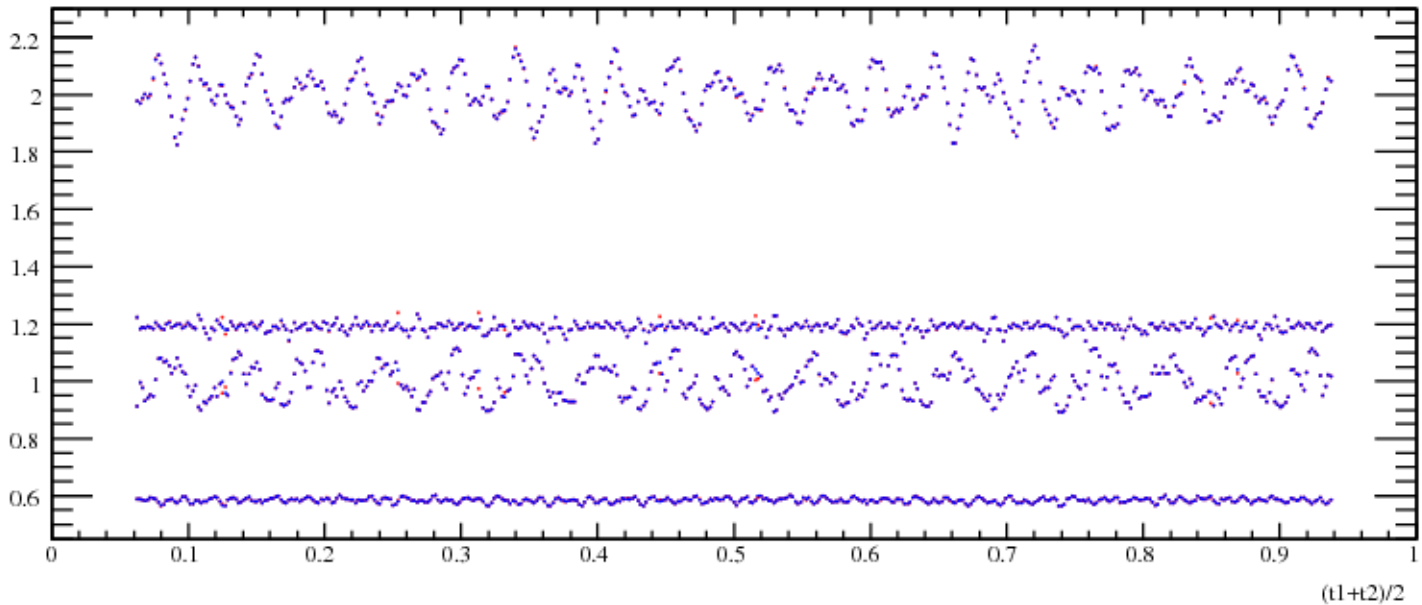
$$F(A, f, \varphi) = \frac{1}{N} \sum_{n=0}^{N-1} \left\{ x(n) - A \cos\left(2\pi \frac{f}{f_s} n + \varphi\right) \right\}^2$$

starting from A and f estimated at 1). This is just a **least square fit with a sinusoidal function**.

- 3) Once the best-fit values of A, f , and ϕ are found, the waveform of converged spectrum is **subtracted from $x(n)$** .
- 4) Repeat the procedure 1 ~ 3 as many times as one would like.

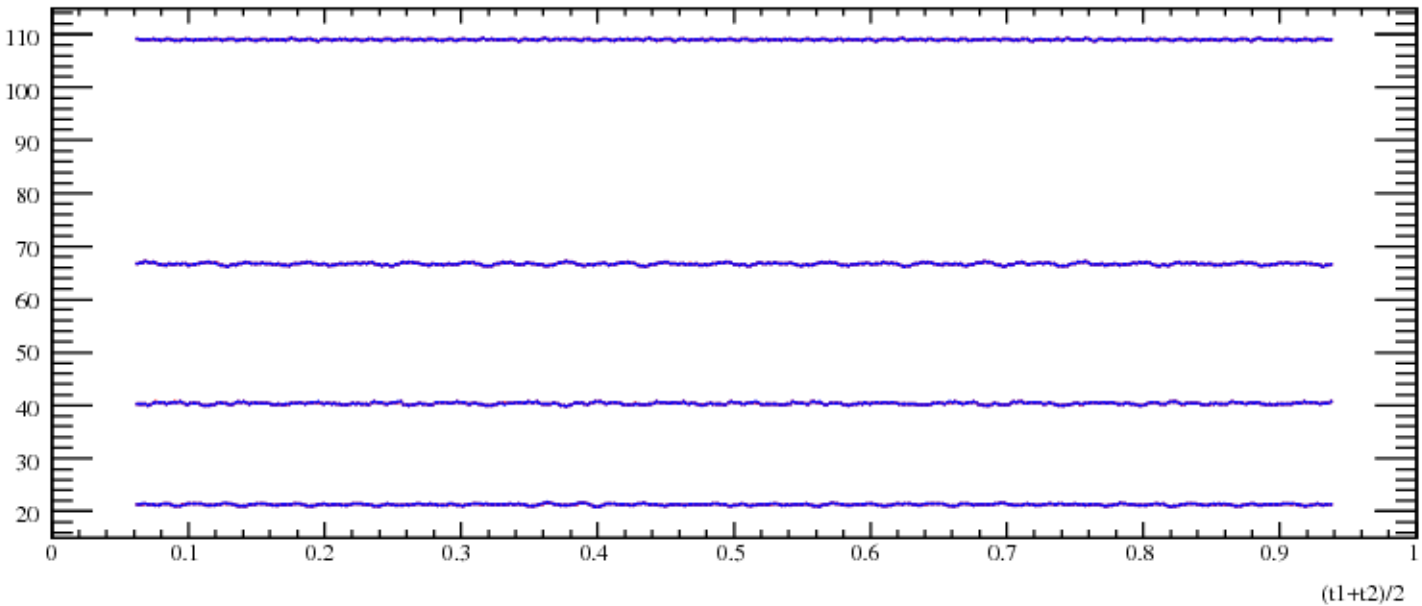
振幅

フレーム長64



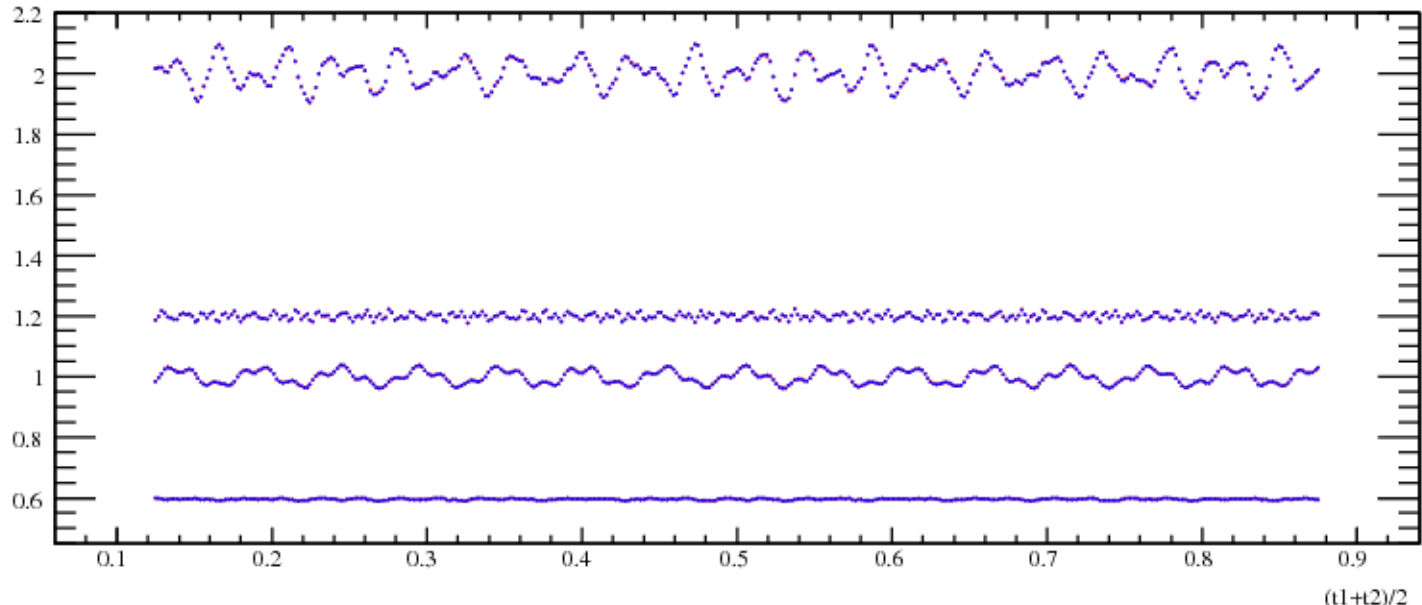
周波数

赤: 富山大
青: 上野

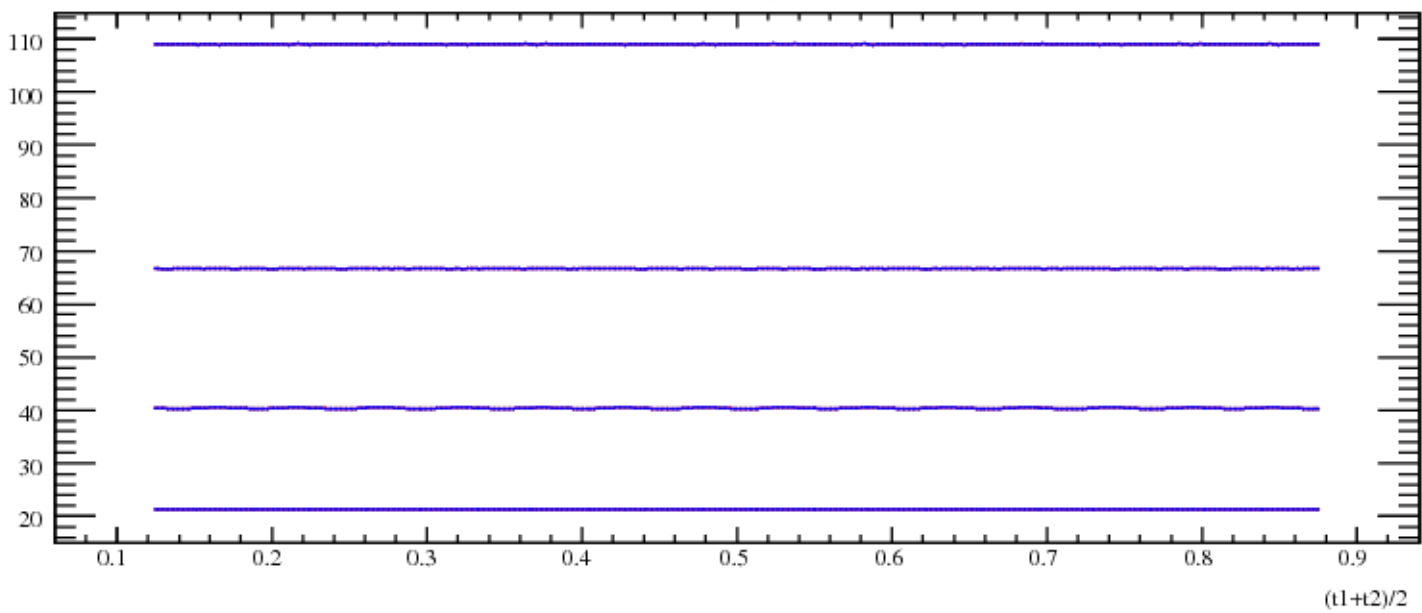


振幅

フレーム長128



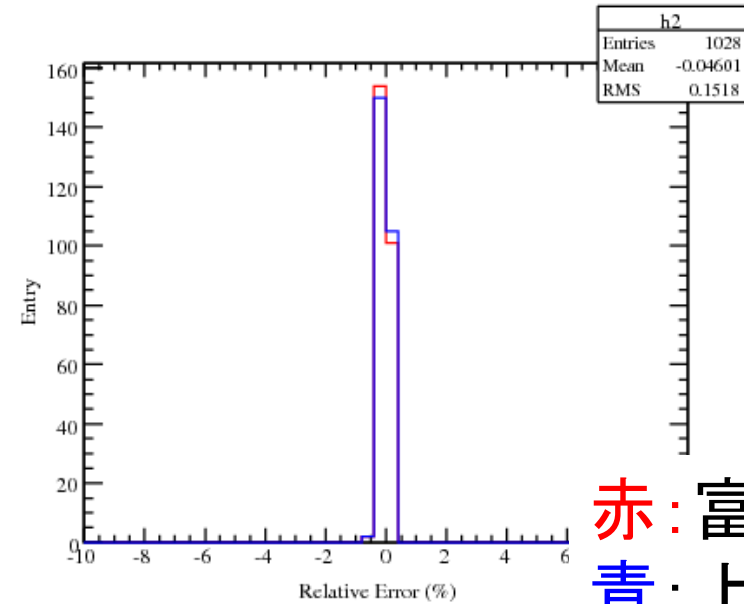
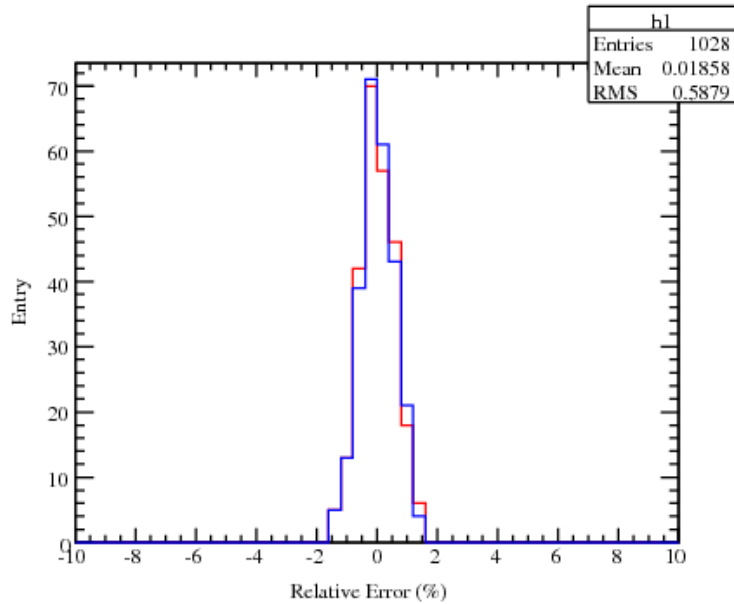
周波数



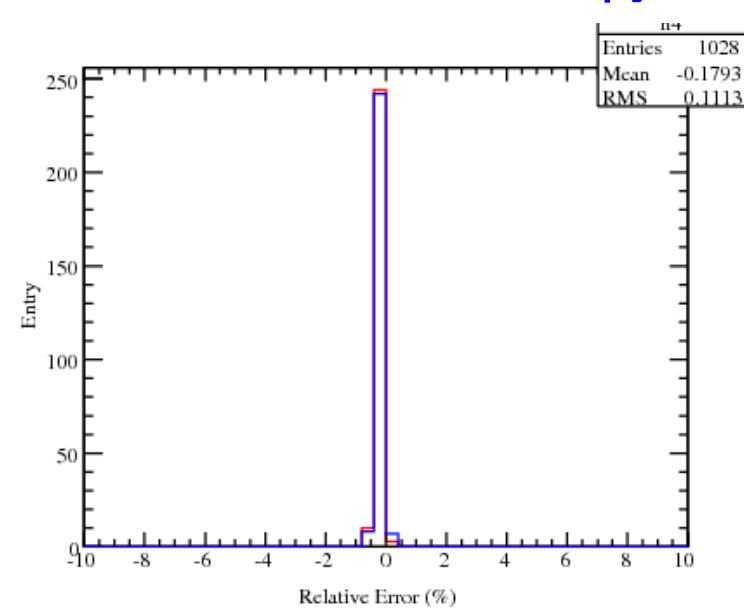
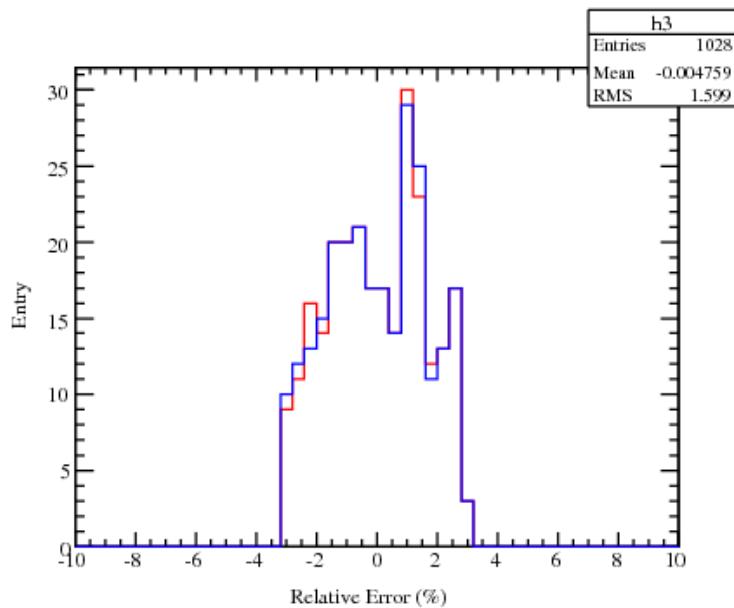
赤: 富山大
青: 上野

振幅分解能

フレーム長256

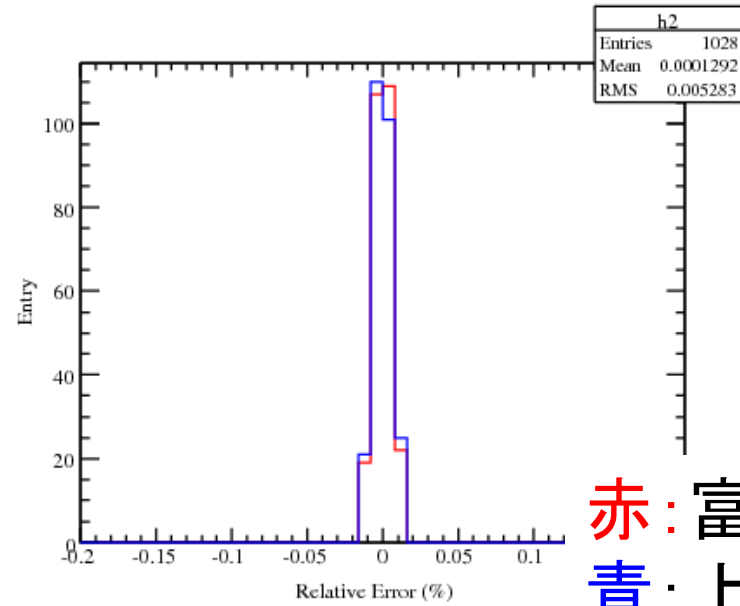
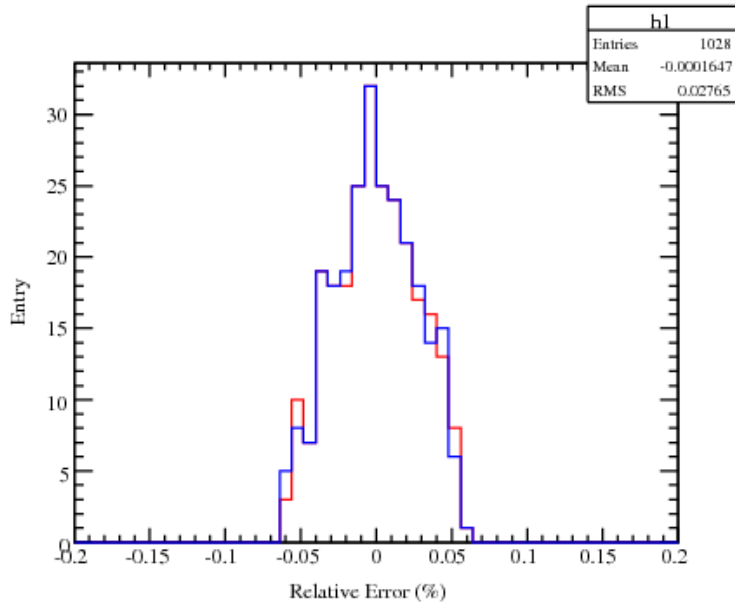


赤: 富山大
青: 上野



周波数分解能

フレーム長256



赤: 富山大
青: 上野

