

ヒ口での常温マルチサンプルノイズ

本原顕太郎

1998 年 9 月 24 日

1 状況

ヒ口を去る直前、ほぼ常温で、Optical Shield のヒーターに電流を流している時と、電流を切っている時にそれぞれマルチサンプルを行なって読み出しの揺らぎがどれくらい少なくなるかを測定した。

2 読み出しノイズ

2.1 データ

いつもの通り、25 枚連続で取得して、不良フレームを捨ててノイズを出した。ただし、読み出しクロックの改良のおかげで不良フレームはほとんど現れなかった。

積分時間は 3 秒～20 秒、検出器の温度は 98/09/17 がほぼ 300K(ちゃんとメモしていなかった)、98/09/18 が 317K だった。

また、98/09/17 はスライダックで Optical Shield に 48V の電圧を加えていた。98/09/18 はスライダックの電源をコンセントから抜いている。

データは以下の通り。

日付	積分時間	N_{sample}	Filename
1998/09/17	3	1	noise.0001.fits ~ noise.0025.fits
	10	1	noise.0026.fits ~ noise.0050.fits
	10	2	noise.0076.fits ~ noise.0100.fits
	10	3	noise.0101.fits ~ noise.0125.fits
	15	4	noise.0126.fits ~ noise.0150.fits
	20	4	noise.0151.fits ~ noise.0175.fits
	20	5	noise.0176.fits ~ noise.0200.fits
	20	6	noise.0201.fits ~ noise.0225.fits
1998/09/18	3	1	noise.0226.fits ~ noise.0250.fits
	15	1	noise.0251.fits ~ noise.0275.fits
	15	2	noise.0276.fits ~ noise.0300.fits
	15	4	noise.0301.fits ~ noise.0325.fits
	25	6	noise.0326.fits ~ noise.0350.fits

2.2 ノイズ

まず、取得したデータそのままで作った stddev フレームの結果を表 1 に示す。しかしこれだけでは、特にマルチサンプルの効果が現れてノイズレベルが低くなると、DC オフセットの影響を除去できていないために、一体どの揺らぎを見ているのかが良くわからなくなってしまうことは N_{Sample} 数が多くなっても揺らぎが少なくなることからも良くわかる。

Frame No.	Quad 1	Quad 2	Quad 3	Quad 4
0001~0025	0.000 $\pm$ 0.0000	4.178 $\pm$ 0.7120	4.136 $\pm$ 0.6573	3.964 $\pm$ 1.4330
0026~0050	0.000 $\pm$ 0.0000	5.168 $\pm$ 1.4480	4.193 $\pm$ 0.6697	5.799 $\pm$ 0.8594
0076~0101	0.000 $\pm$ 0.0000	2.635 $\pm$ 0.4355	2.835 $\pm$ 0.4505	3.022 $\pm$ 0.5766
0101~0125	0.000 $\pm$ 0.0000	1.894 $\pm$ 0.3618	2.172 $\pm$ 0.4384	2.070 $\pm$ 0.3962
0126~0150	0.000 $\pm$ 0.0000	1.705 $\pm$ 0.3018	1.981 $\pm$ 0.3658	1.890 $\pm$ 0.3237
0151~0175	0.000 $\pm$ 0.0000	1.989 $\pm$ 0.3237	2.228 $\pm$ 0.3807	2.080 $\pm$ 0.3759
0176~0200	0.000 $\pm$ 0.0000	1.616 $\pm$ 0.2766	1.572 $\pm$ 0.3227	1.747 $\pm$ 0.2911
0201~0225	0.000 $\pm$ 0.0000	1.430 $\pm$ 0.2347	1.516 $\pm$ 0.3196	1.640 $\pm$ 0.2692
0226~0250	0.000 $\pm$ 0.0000	2.643 $\pm$ 0.4853	2.899 $\pm$ 0.5296	2.777 $\pm$ 0.4849
0251~0275	0.000 $\pm$ 0.0000	3.388 $\pm$ 0.5210	3.721 $\pm$ 0.5975	3.967 $\pm$ 0.5835
0276~0300	0.000 $\pm$ 0.0000	2.266 $\pm$ 0.3723	2.647 $\pm$ 0.4339	2.629 $\pm$ 0.4003
0301~0325	0.000 $\pm$ 0.0000	1.618 $\pm$ 0.3103	1.707 $\pm$ 0.3502	1.813 $\pm$ 0.3371
0326~0350	0.000 $\pm$ 0.0000	1.739 $\pm$ 0.2652	1.986 $\pm$ 0.3343	1.960 $\pm$ 0.2753

表 1: 生フレームのままで作った stddev フレームの imstat の結果。値は mode、誤差は stddev である。単位は ADU r.m.s.。

Frame No.	Quad 1	Quad 2	Quad 3	Quad 4
0001~0025	0.000 $\pm$ 0.0000	3.326 $\pm$ 0.006	3.445 $\pm$ 0.001	3.414 $\pm$ 0.034
0026~0050	0.000 $\pm$ 0.0000	3.351 $\pm$ 0.045	3.423 $\pm$ 0.001	3.240 $\pm$ 0.037
0076~0101	0.000 $\pm$ 0.0000	2.595 $\pm$ 0.066	2.722 $\pm$ 0.001	2.542 $\pm$ 0.002
0101~0125	0.000 $\pm$ 0.0000	1.955 $\pm$ 0.008	2.138 $\pm$ 0.001	1.996 $\pm$ 0.025
0126~0150	0.000 $\pm$ 0.0000	1.665 $\pm$ 0.001	1.848 $\pm$ 0.000	1.677 $\pm$ 0.004
0151~0175	0.000 $\pm$ 0.0000	1.662 $\pm$ 0.008	1.842 $\pm$ 0.006	1.676 $\pm$ 0.015
0176~0200	0.000 $\pm$ 0.0000	1.461 $\pm$ 0.000	1.633 $\pm$ 0.000	1.460 $\pm$ 0.001
0201~0225	0.000 $\pm$ 0.0000	1.328 $\pm$ 0.000	1.489 $\pm$ 0.000	1.324 $\pm$ 0.000
0226~0250	0.000 $\pm$ 0.0000	2.575 $\pm$ 0.005	2.836 $\pm$ 0.003	2.581 $\pm$ 0.006
0251~0275	0.000 $\pm$ 0.0000	2.598 $\pm$ 0.008	2.865 $\pm$ 0.006	2.506 $\pm$ 0.071
0276~0300	0.000 $\pm$ 0.0000	1.916 $\pm$ 0.002	2.165 $\pm$ 0.002	1.926 $\pm$ 0.004
0301~0325	0.000 $\pm$ 0.0000	1.404 $\pm$ 0.001	1.594 $\pm$ 0.001	1.423 $\pm$ 0.002
0326~0350	0.000 $\pm$ 0.0000	1.224 $\pm$ 0.010	1.382 $\pm$ 0.010	1.241 $\pm$ 0.011

表 2: 続きのフレームの差をとって求めた読み出しノイズ。誤差は stddev である。単位は ADU r.m.s.。

また、常温でのデータ取得のため、例の縞縞が現れてしまって、局面でフィットして DC オフセットを除去する方法もうまくいかないことがわかった。そこで、以下の方法でノイズを定義した。

1. 番号が続きの 2 フレームの差をとる
2. 差のフレームの各コドラントのピクセル値の stddev を
imstat (image name) upper=10 lower=-10
で出す
3. これを各組 24 フレームについて行ない、その平均を読み出しノイズとした

結果を表 2 に示す。これを見ると、マルチサンプルを行なうほどに読み出しノイズが小さくなっているのが良くわかる。とくに、6 回のマルチサンプルで読み出しノイズが 1.2 ADU rms (3.6 e^- rms 相当) というのは驚くべき数字だった。

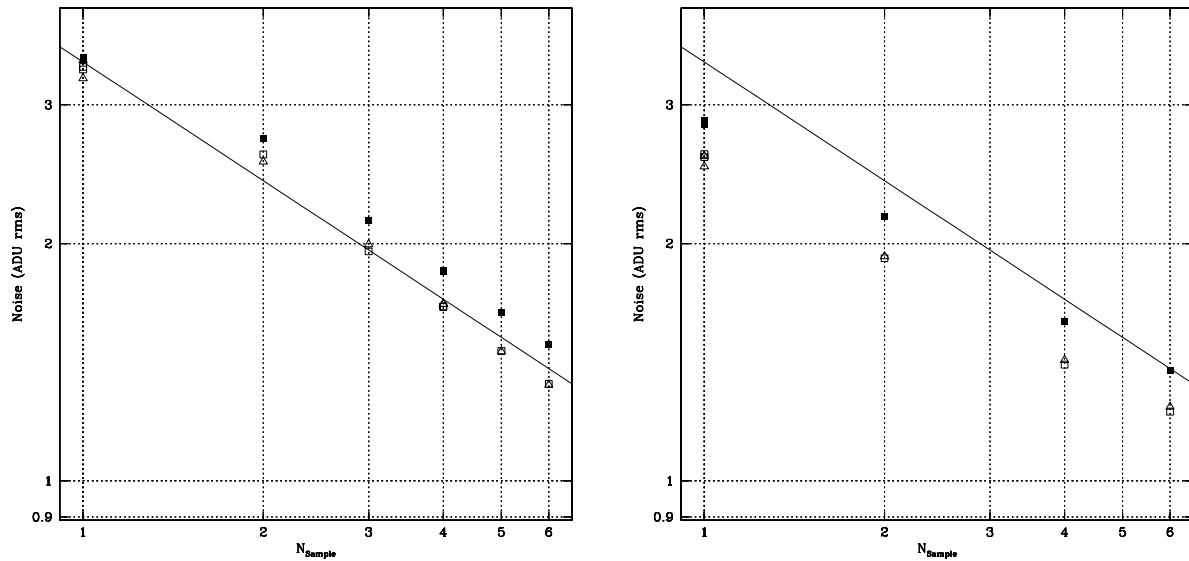


図 1: 読み出しノイズと N_{Sample} の関係。左が 98/09/17、右が 98/09/18 の結果。白四角、黒四角、白三角の順にコドラント 2, 3, 4 を表している。グラフ内の実線は $Noise = \sqrt{N_{sample}}$ 。

果たしてこの値が冷却した状態で再現されるかが興味深いところである。

さらに、読み出しノイズと N_{Sample} をプロットしたグラフを図 1 に示す。

これを見ると、 N_{Sample} を増やしていくと読み出しノイズはほぼ自乗則に乗って減っていくのがわかる。ただ、いくつかポイントがある。

- 98/09/18の方が読み出しノイズが小さい。これは、ヒーターの電源線回りからノイズが混入していた可能性を示唆している。
- 日によって、傾きが違っている。ヒーターを止めてコンセントを抜いた方が N_{Sample} を増やしてもノイズの減少の割合が少ないことになる。つまり、ヒーターの電源線回りから回り込んでいるノイズ成分はマルチサンプルで落ちるが、残りの成分は何らかの周期成分が混ざり込んでいて、マルチサンプルでは完全に落とし切れていないと解釈するのがいまのところもっとも自然であろう。
- コドラント 3 だけ、ノイズが少し大きい。これは以前から出ていた現象である。

2.3 フレームの安定性

次に、二次元化したフレームを図 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 に示す。

安定性は両日でそんなに変わらない。また、マルチサンプルの回数を増やすにつれてギザギザの幅が小さくなっていくのが明らかにわかる。

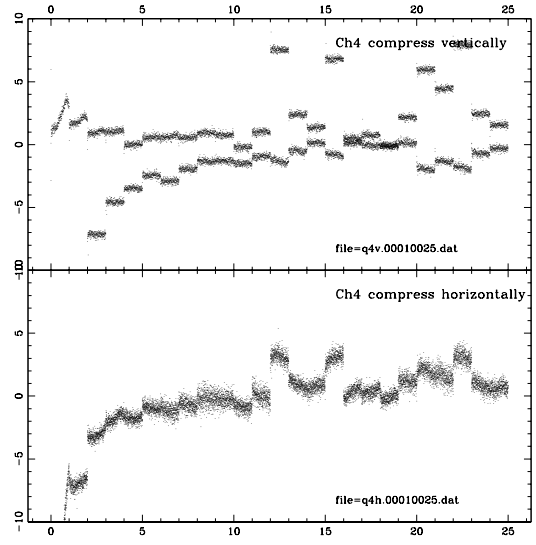
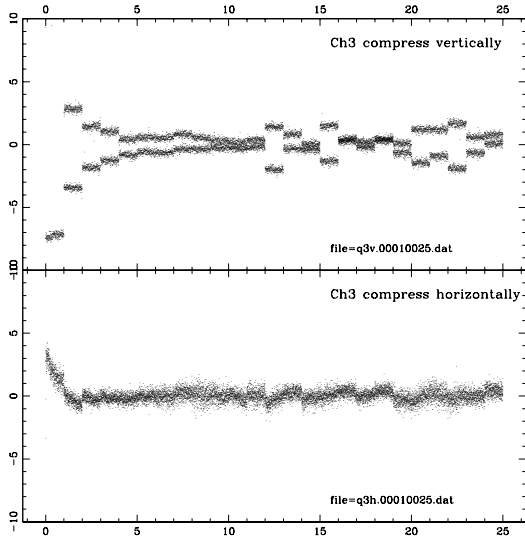
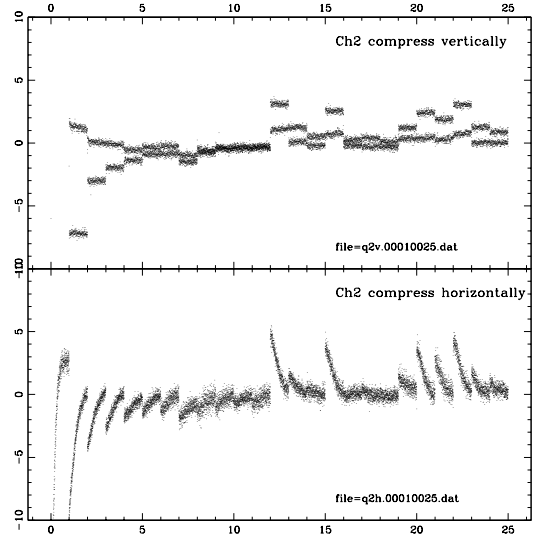
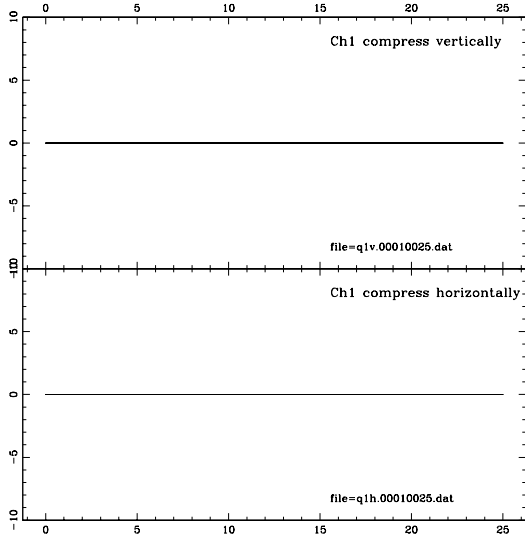


Figure 2: noise.0001.fits $\sim$ noise.0025.fits

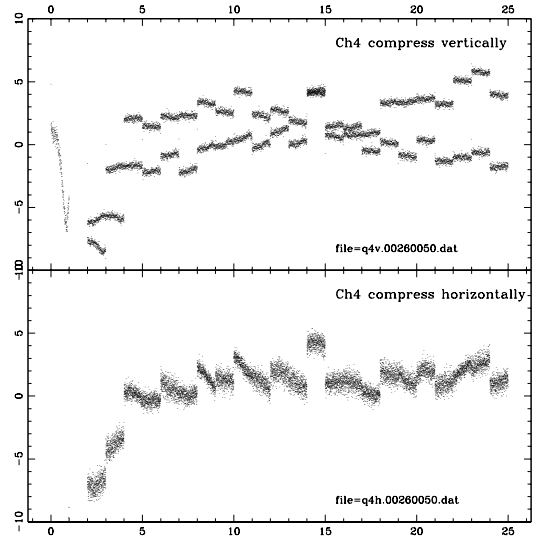
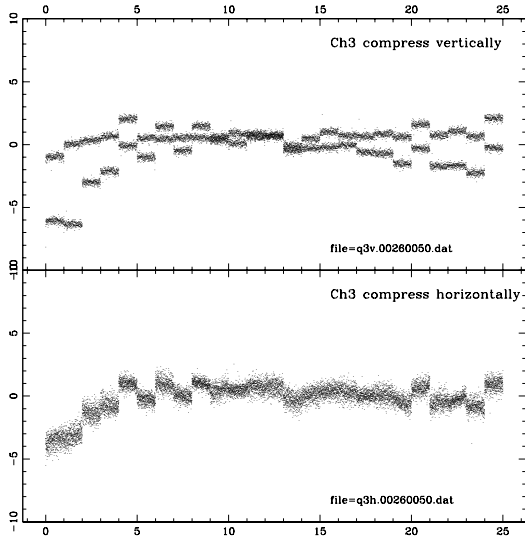
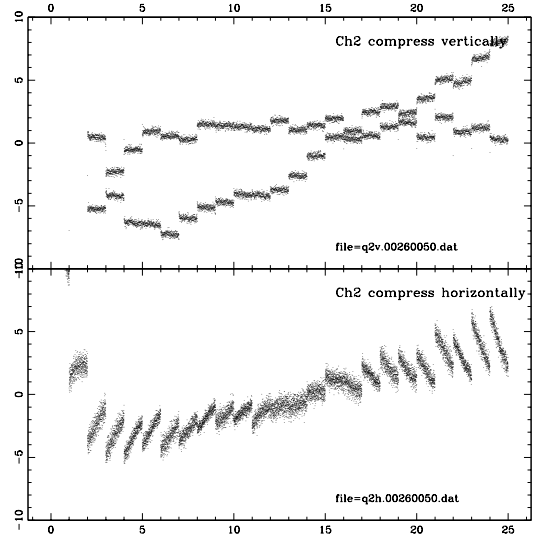
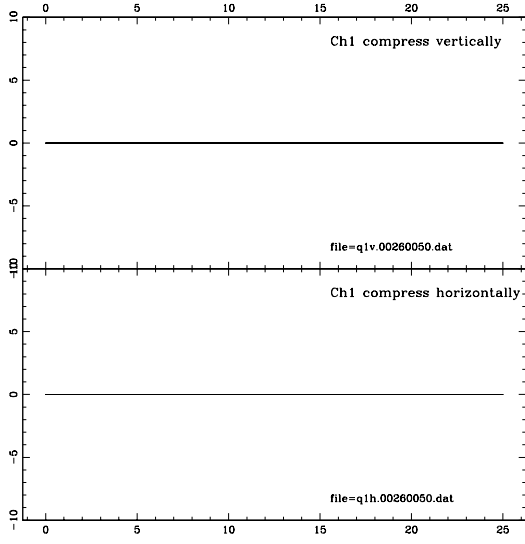


Figure 3: noise.0026.fits $\sim$ noise.0050.fits

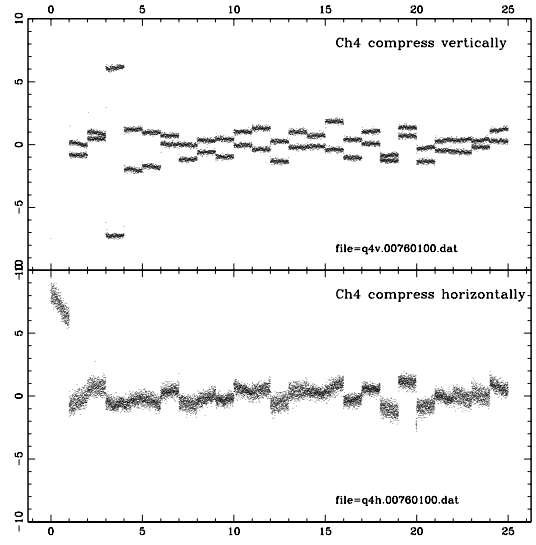
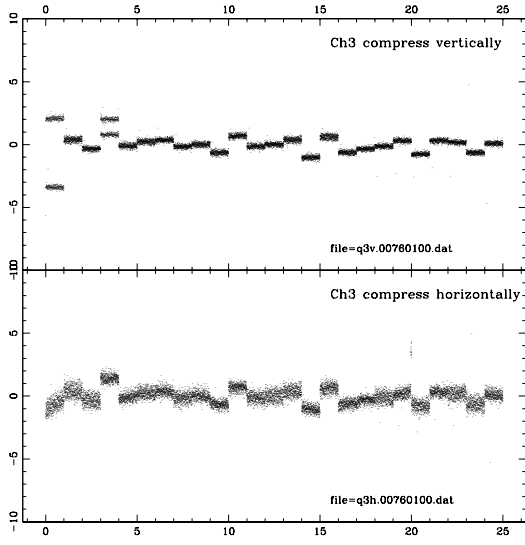
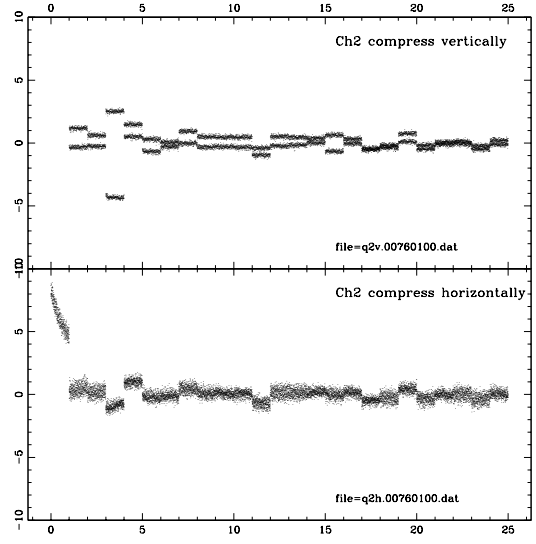
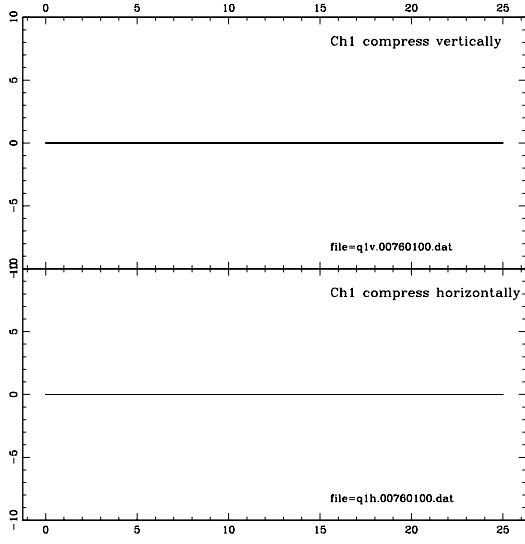


图 4: noise.0076.fits $\sim$ noise.0100.fits

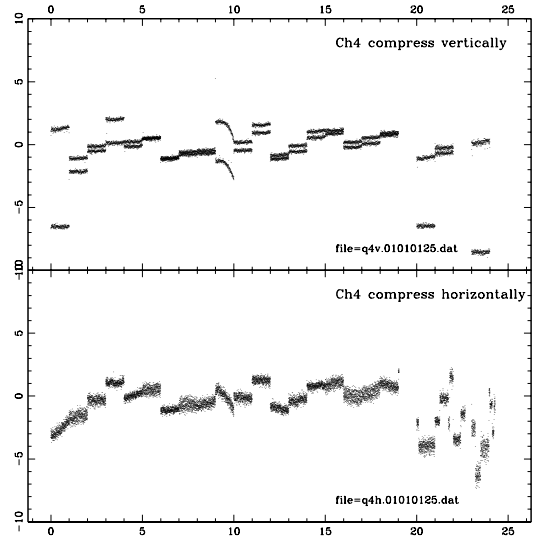
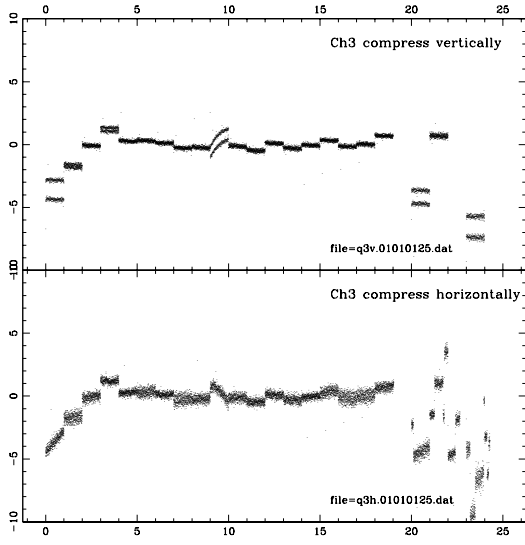
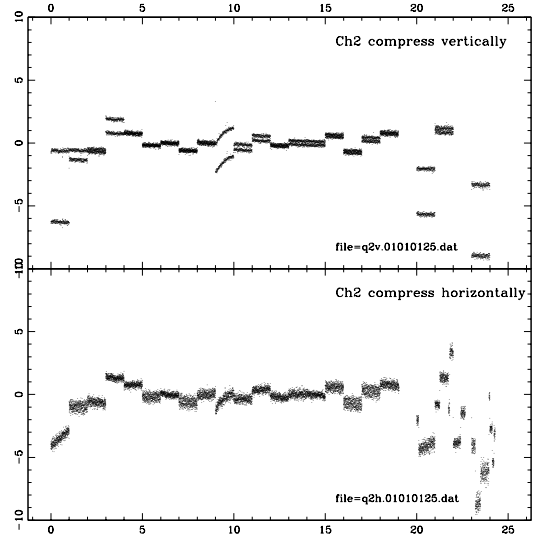
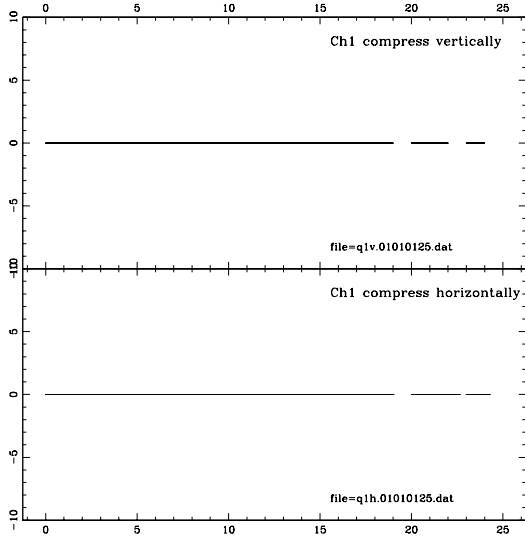


图 5: noise.0101.fits $\sim$ noise.0125.fits

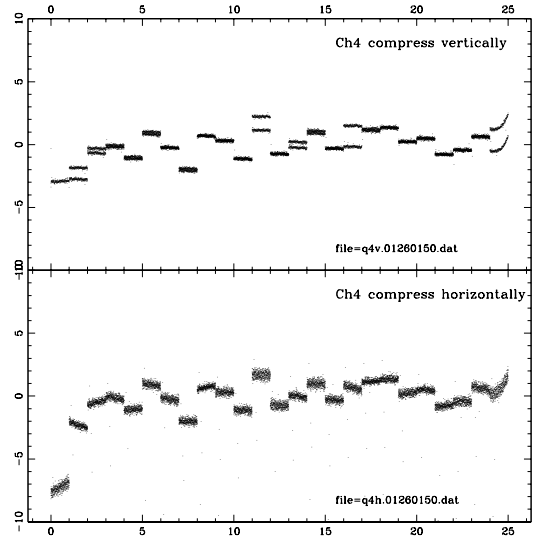
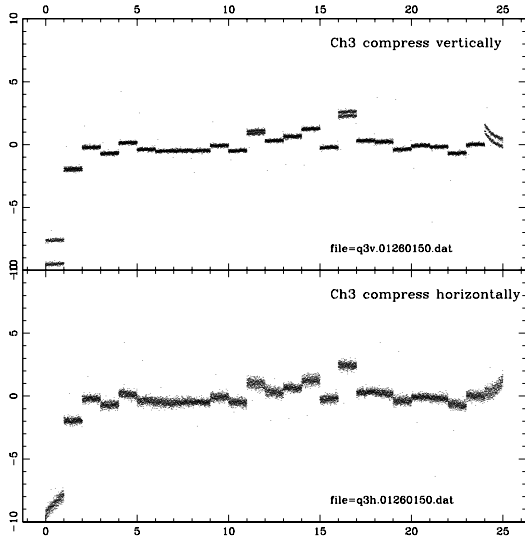
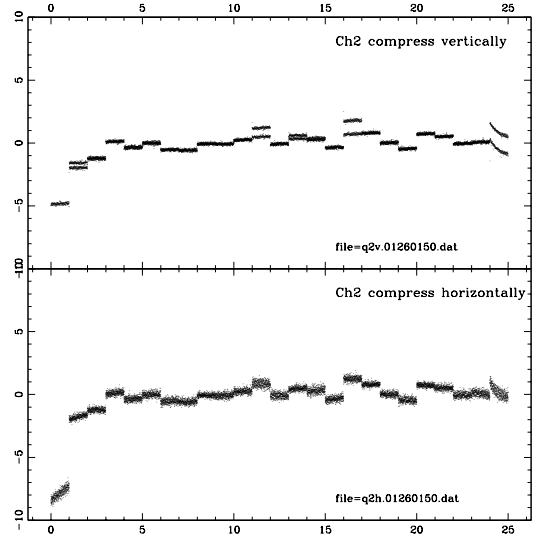
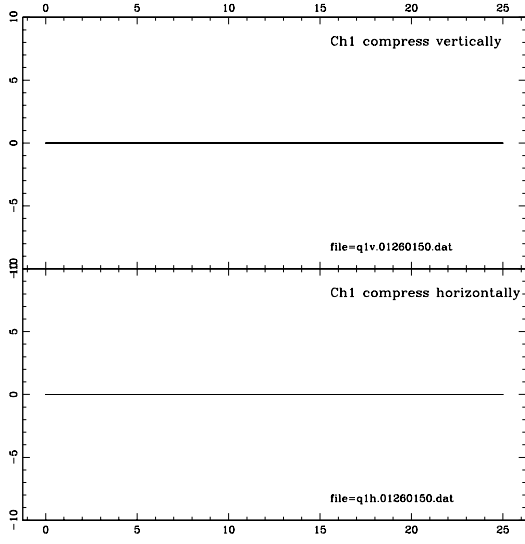


Figure 6: noise.0126.fits $\sim$ noise.0150.fits

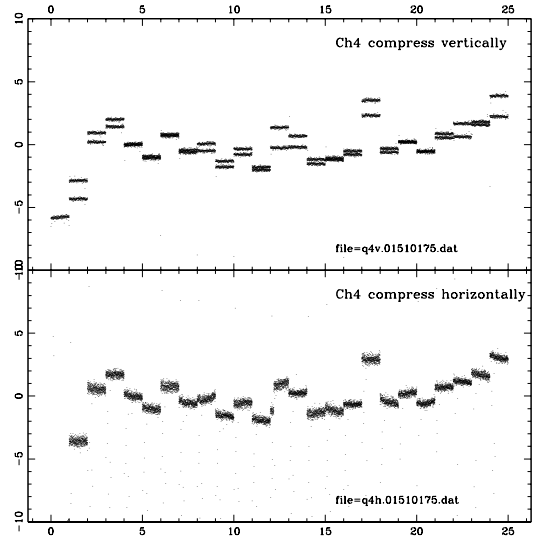
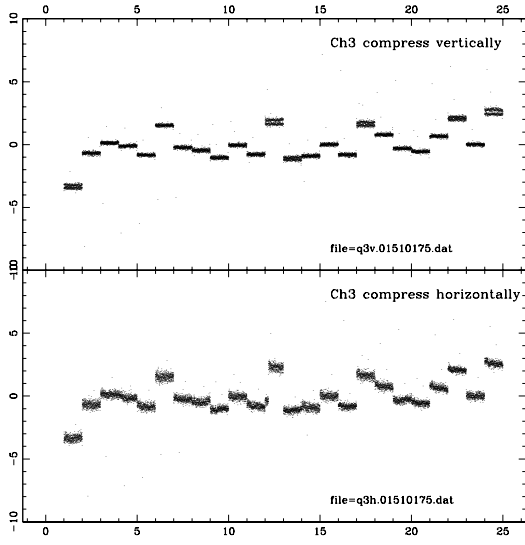
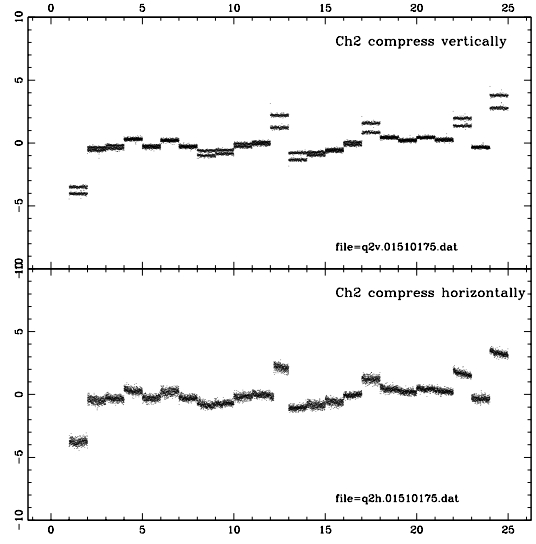
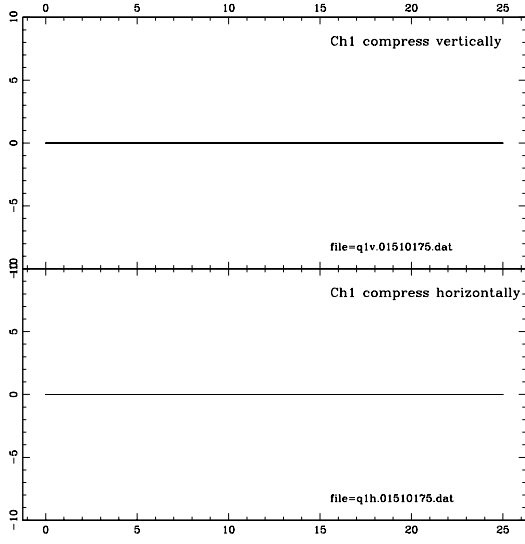


图 7: noise.0151.fits $\sim$ noise.0175.fits

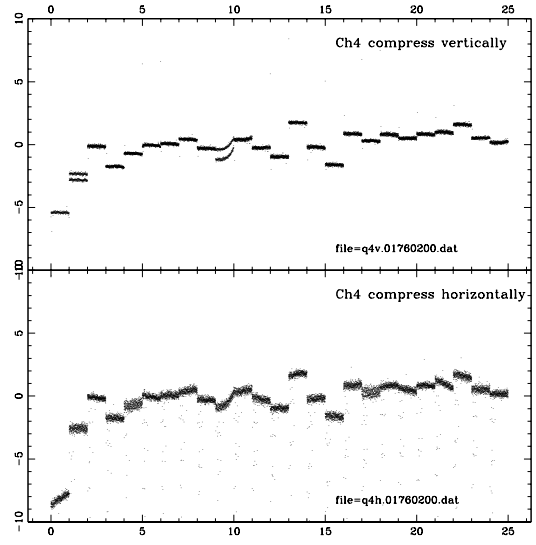
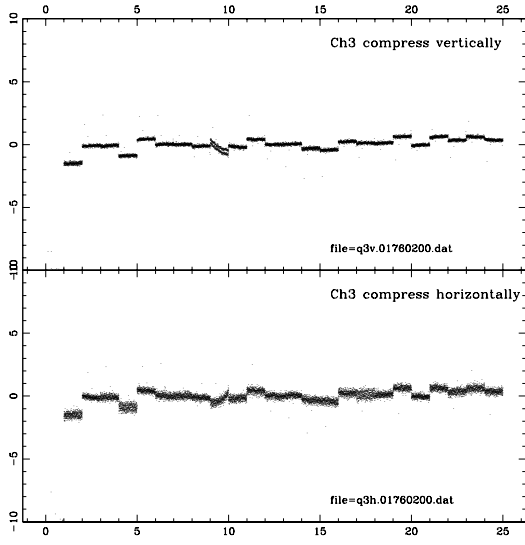
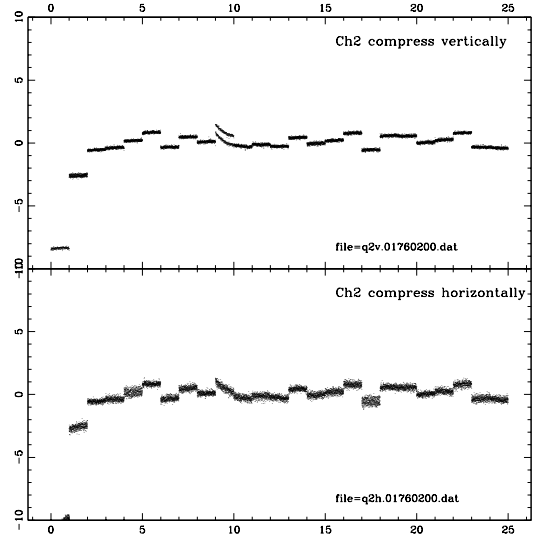
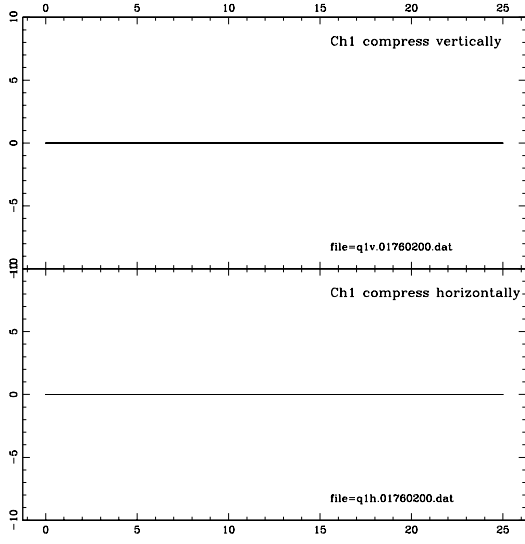


Fig 8: noise.0176.fits $\sim$ noise.0200.fits

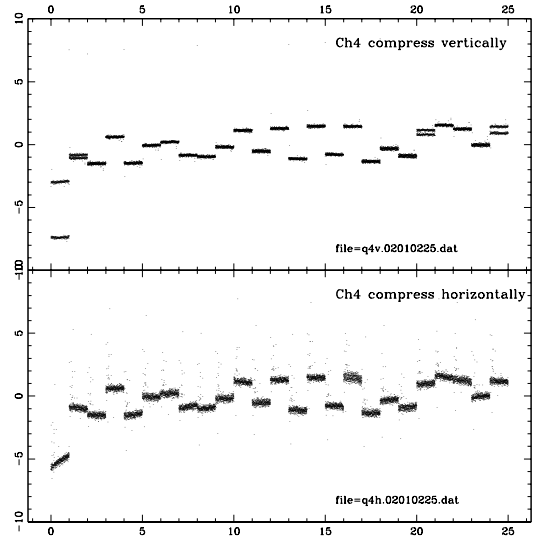
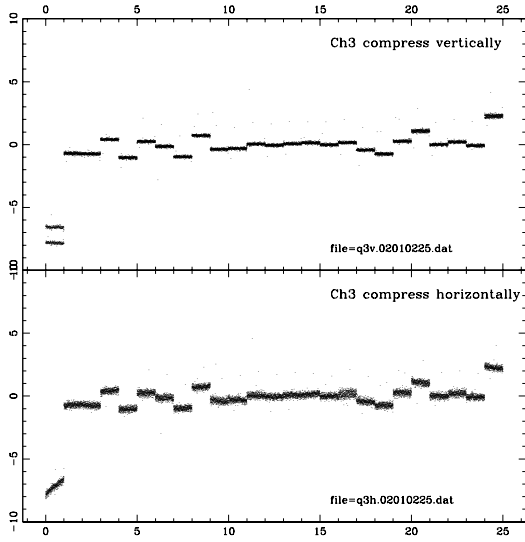
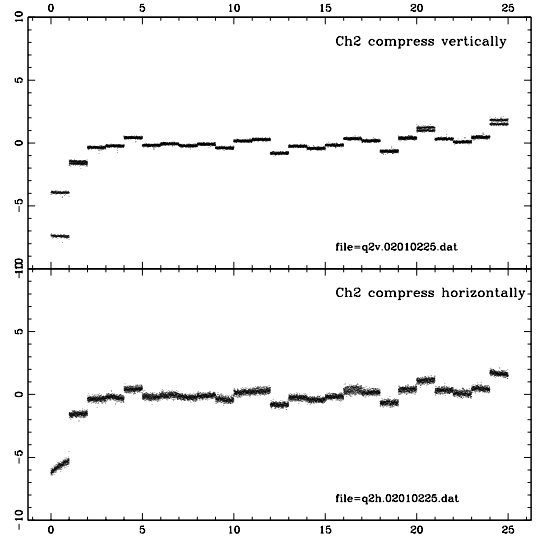
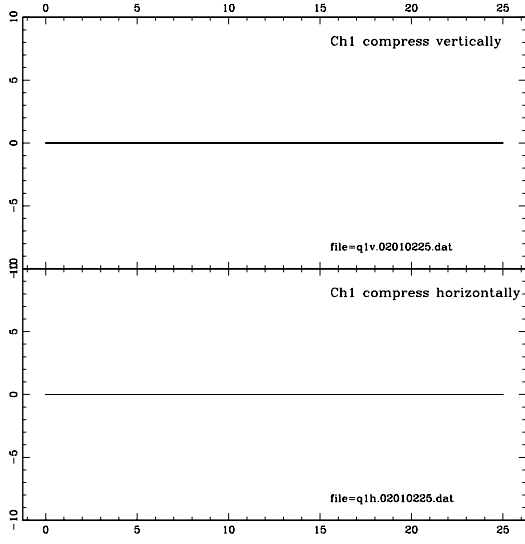


图 9: noise.0201.fits $\sim$ noise.0225.fits

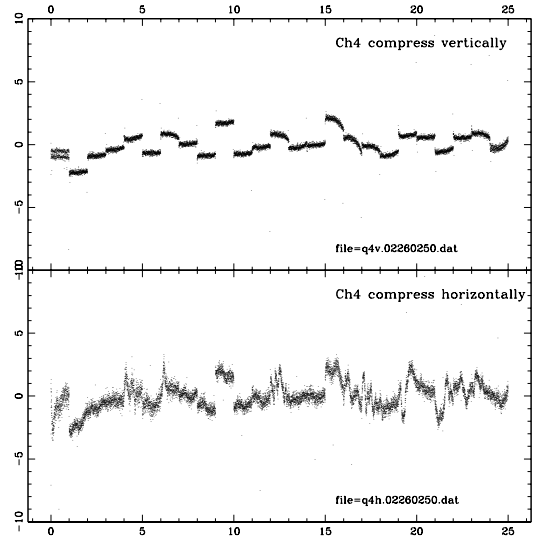
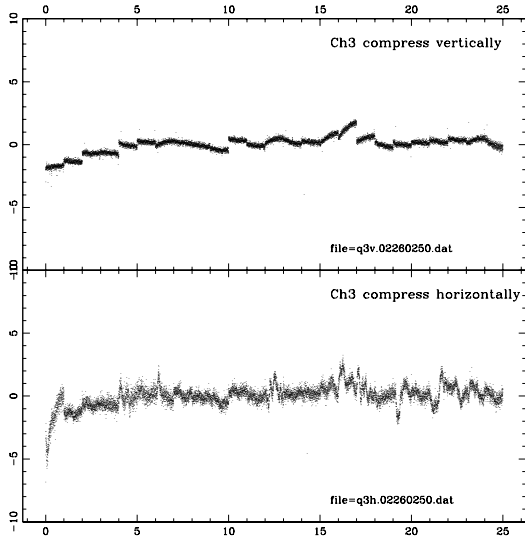
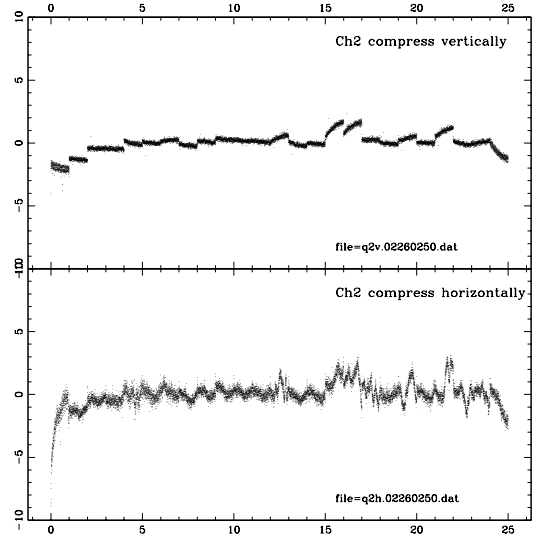
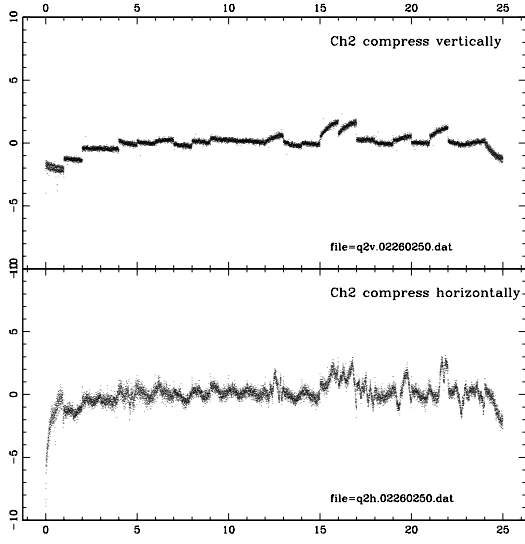


Figure 10: noise.0226.fits $\sim$ noise.0250.fits

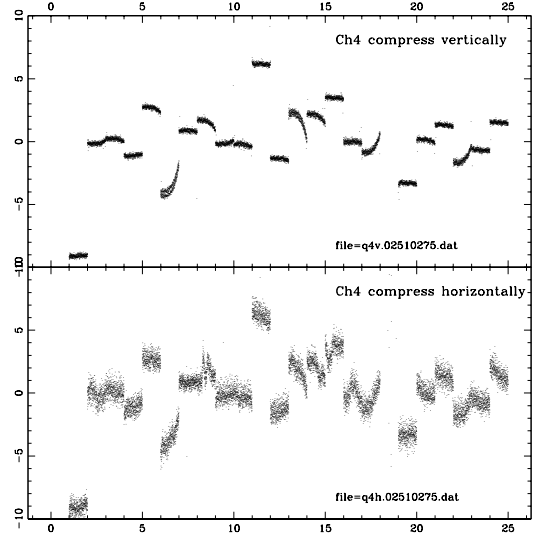
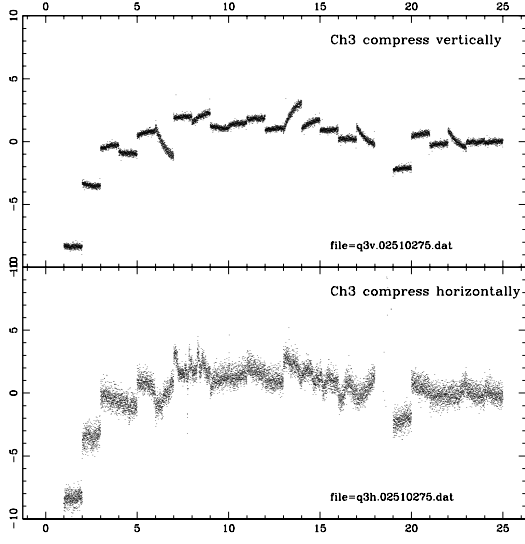
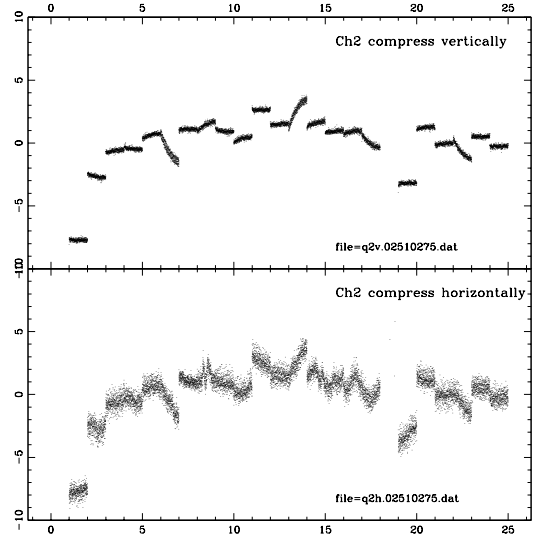
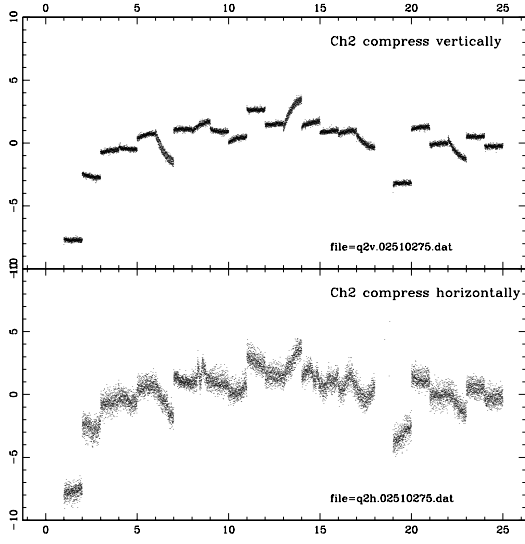


Fig 11: noise.0251.fits $\sim$ noise.0275.fits

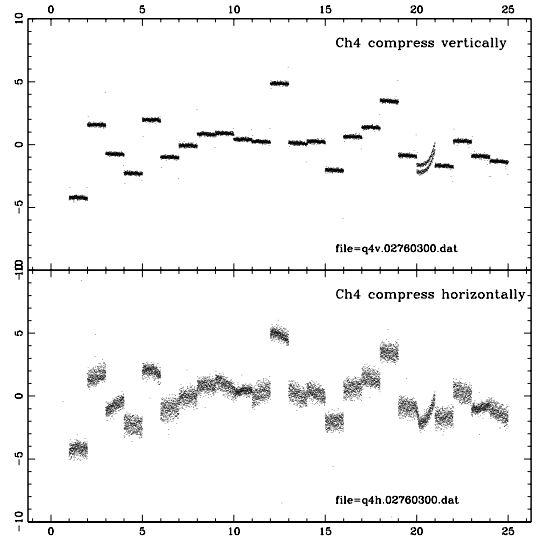
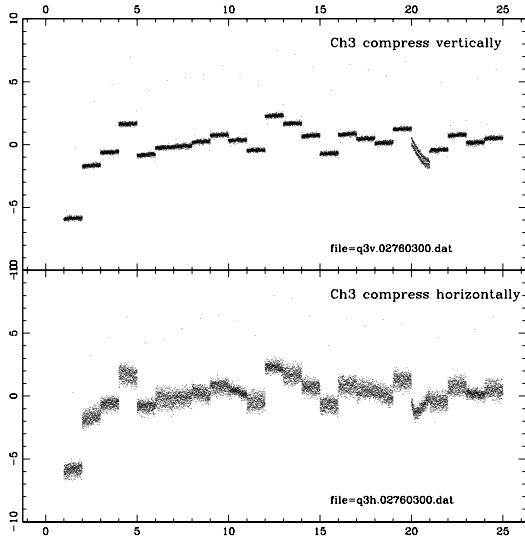
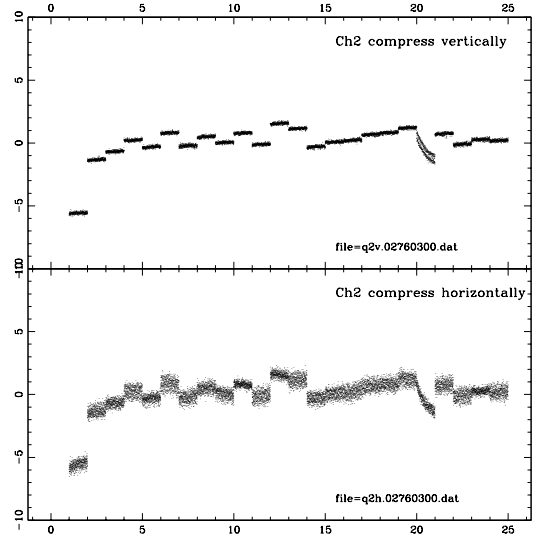
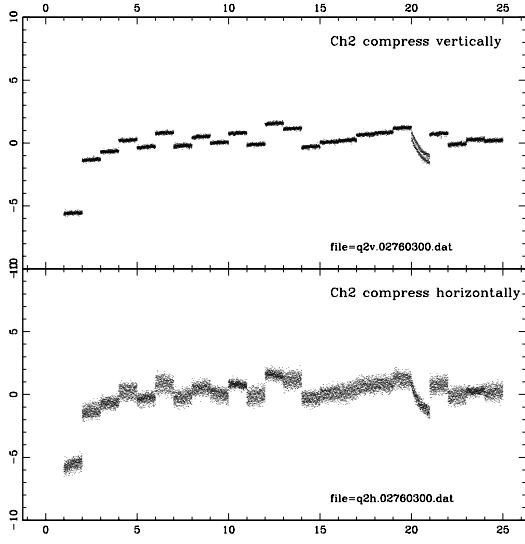


Figure 12: noise.0276.fits $\sim$ noise.0300.fits

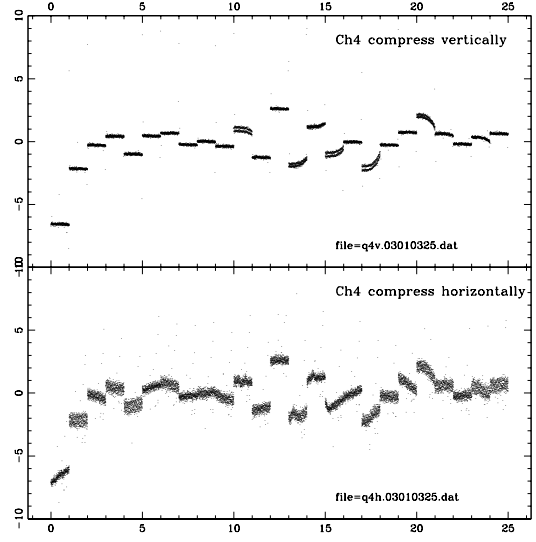
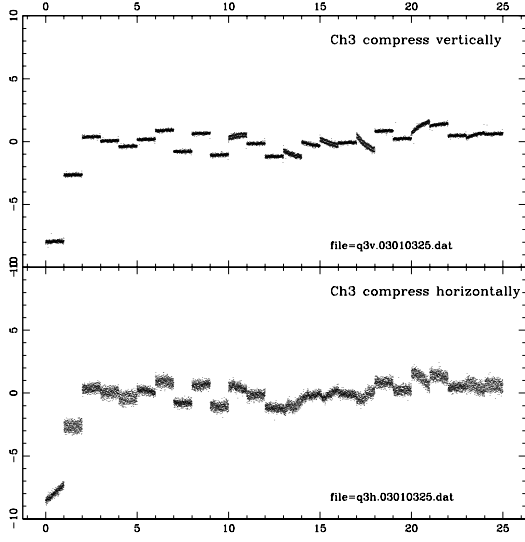
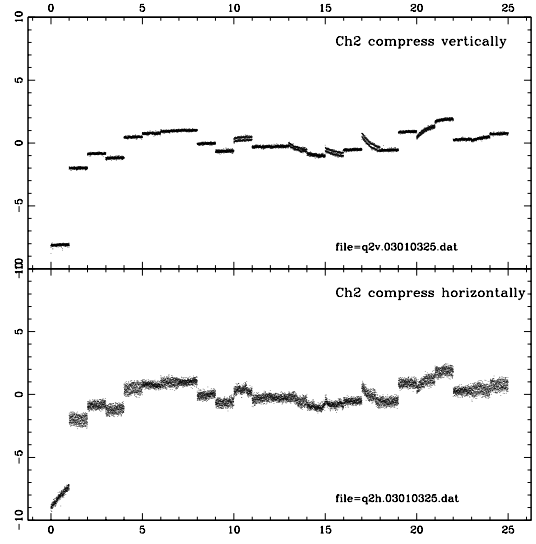
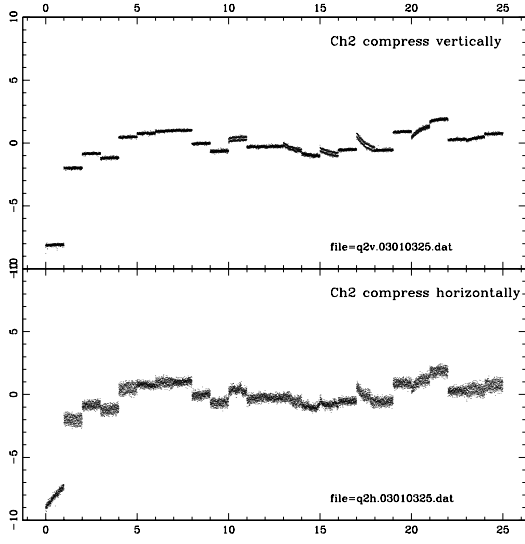


Figure 13: noise.0301.fits $\sim$ noise.0325.fits

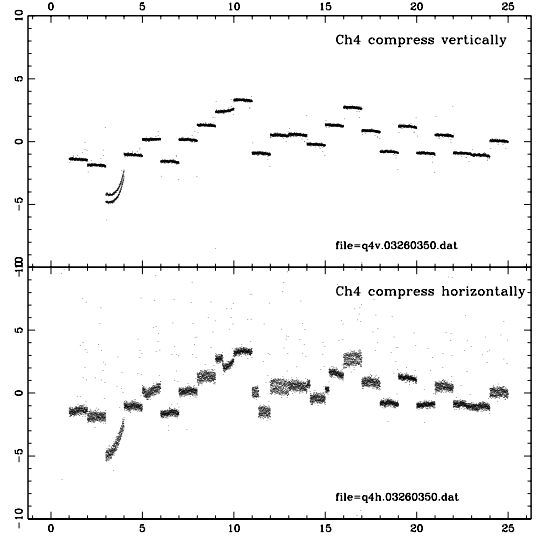
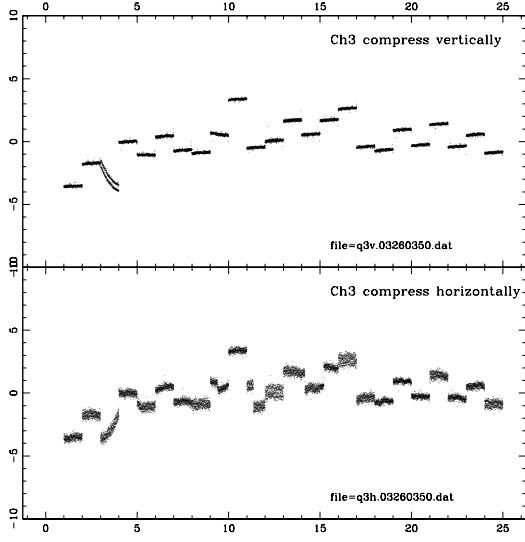
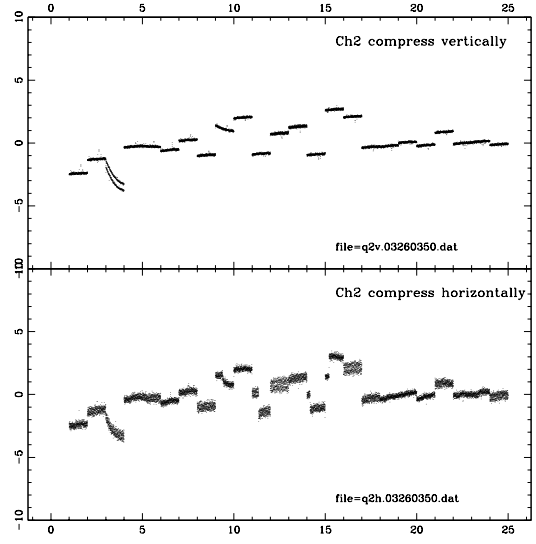
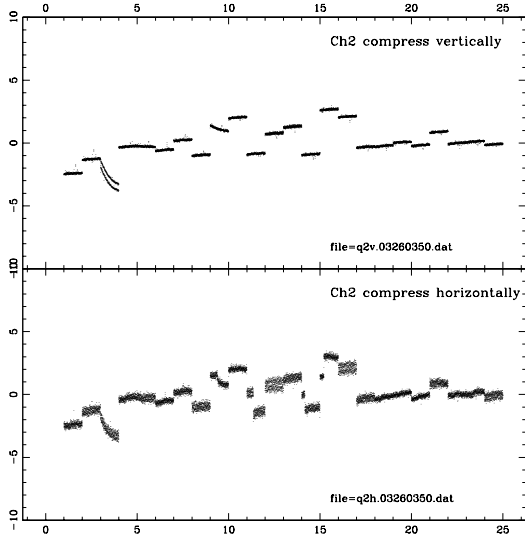


Figure 14: noise.0326.fits $\sim$ noise.0350.fits