GIS PH モード 時刻ビット 0 での時刻付けについて

平山 昌治

1995 年 9 月 1 日

本レポートでは、PSR0540–69 の観測データを用いて、GIS PH モード 時刻ビット 0 における時刻付けの信頼性について考察する。

1 GIS の時刻付けの原理

解析方法と結果の報告の前に、簡単に GIS PH モードでの時刻付けの方法について説明しておく。詳細については、石崎 M 論 [1] や中間報告書 [2] などを参照するとよい。参考のために、図 1にオンボードでの処理の手順を、時刻付けに関連のある部分だけ摸式的に示す。

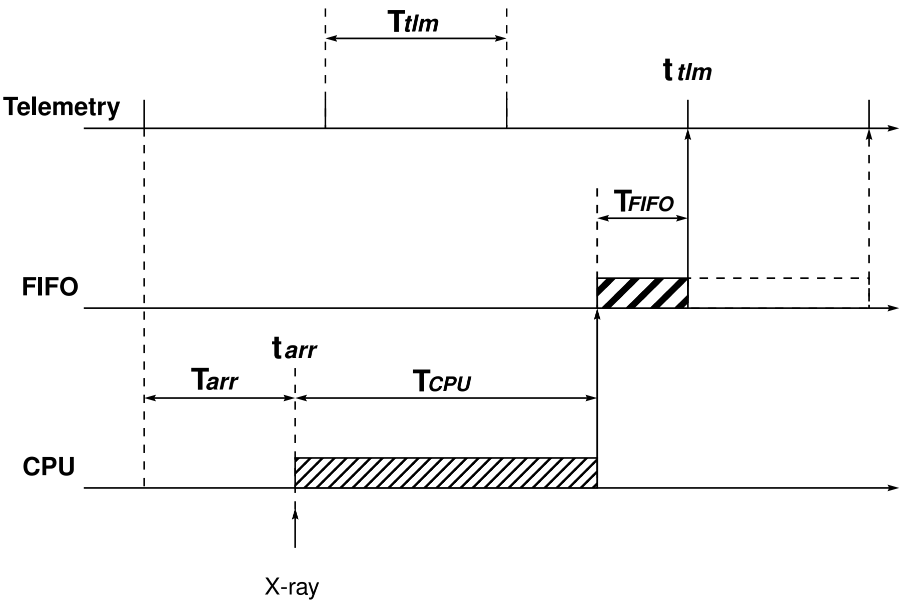

図 1: X 線が検出されてからそのイベントがテレメトリに出力されるまでの概念図。図中、テレメトリ編集時刻は “Telemetry” 軸上の tick で表され、“CPU” 軸上に “X-ray” と印の打ってある時刻に X 線イベントが検出されたとする。各記号の意味は、それぞれ、t_{arr}: X 線が検出された時刻、t_{tlm}: 検出された X 線イベントがテレメトリに出力された時刻、T_{tlm}: テレメトリ編集間隔、T_{arr}: X 線が検出された時刻に最も近いテレメトリ編集時刻から X 線検出時刻までの時間、T_{CPU}: CPU がテレメトリに出力するデータを収集、計算するのにかかる時間、T_{FIFO}: X 線イベントが FIFO に詰められてからテレメトリに出力されるまでの時間である。

まず、GIS で X 線イベントが検出されると、光子到着時刻を記録するためのカウンタがラッチされ、X 線イベントが検出されたことを示すフラグが立つ。以下では、このカウンタの値をもとにつけられる時刻を arrival time t_{arr}と呼ぶことにする。CPU は X 線検出を示すフラグを見つけると、イベント情報 (パルスハイトや検出位置) を収集、計算し、結果を FIFO に詰める。このとき、時刻ビットが “0” でない観測の時には、光子到着時刻記録用のカウンタの値も同時に収集される。テレメトリ編集タイミング

がやってくると、CPU は FIFO の中の一番古いイベントをテレメトリに出力する。以下では、この時刻を telemetry time t_{tlm} と呼ぶ。

テレメトリに出力されたイベントは地上へ転送され、地上局では、各々のフレームに対して、KSC の原子時計を用いて UTC に較正された時刻が付けられる。したがって、t_{tlm} はいかなる観測でも計算することのできる時刻である。さらに、時刻ビットが "0" でない観測では、t_{tlm} と光子到着時刻記録用のカウンタの値を用いて t_{arr} が推定され、イベント時刻とされる。時刻ビットが "0" の観測の場合には t_{arr} が推定できないので、イベント時刻としては t_{tlm} が用いられる。

2 t_{arr} と t_{tlm} のずれ

GIS で X 線が検出されてからテレメトリに出力されるまでの処理は、概ね、図 1のようになっている。図 1では、テレメトリ編集時刻は "Telemetry" 軸上の tick で表され、図 1中の記号の意味は、

t_{arr}	X 線が検出された時刻
t_{tlm}	検出された X 線イベントがテレメトリに出力された時刻
T_{tlm}	テレメトリ編集間隔
T_{arr}	X 線が検出された時刻に最も近いテレメトリ編集時刻から X 線検出時刻までの時間
T_{CPU}	CPU がテレメトリに出力するデータを収集、計算するのにかかる時間
T_{FIFO}	X 線イベントが FIFO に詰められてからテレメトリに出力されるまでの時間

である。このとき、図 1から明らかに、

$$T_{arr} + T_{CPU} + T_{FIFO} = n \times T_{tlm}$$

の関係がある。ただし、

$$n = \begin{cases} \left[\frac{T_{CPU}}{T_{tlm}}\right] + 1 + N_{FIFO} & \frac{T_{arr}+T_{CPU}}{T_{tlm}} \le \left[\frac{T_{CPU}}{T_{tlm}}\right] + 1 \text{ のとき} \\ \left[\frac{T_{CPU}}{T_{tlm}}\right] + 2 + N_{FIFO} & \text{それ以外のとき} \end{cases}$$

で、N_{FIFO} は FIFO に詰め込む時点で先に FIFO 内に入っているイベントの数[1]である。また、$[x]$ は「x を超えない最大の整数」を表す (Gauss の記号)。したがって、X 線が検出されてからそのイベントがテレメトリに出力されるまでにかかる時間 T_{delay} は、$N_{FIFO} = 0$ のとき、

$$\begin{aligned} T_{delay} &\equiv t_{tlm} - t_{arr} \\ &= T_{CPU} + T_{FIFO} \\ &= n \times T_{tlm} - T_{arr} \\ &= \begin{cases} \left[\frac{T_{CPU}}{T_{tlm}}\right] T_{tlm} + T_{tlm} - T_{arr} & \frac{T_{arr}+T_{CPU}}{T_{tlm}} \le \left[\frac{T_{CPU}}{T_{tlm}}\right] + 1 \text{ のとき} \\ \left[\frac{T_{CPU}}{T_{tlm}}\right] T_{tlm} + 2T_{tlm} - T_{arr} & \text{それ以外のとき} \end{cases} \end{aligned}$$

と表される。$N_{FIFO} \neq 0$ のときは、これに、$N_{FIFO} \times T_{tlm}$ (定数) が加算される。

さて、T_{tlm} と T_{CPU} が定数の場合には、T_{delay} と T_{arr} の関係は図 2のようになる ($N_{FIFO} = 0$ のとき)。X 線の検出がテレメトリ編集時刻とは無関係にランダムに起こる場合[2]には、T_{arr} は $0 \le T_{arr} < T_{tlm}$ なる区間に一様に分布する。このとき、図 2により、T_{delay} は $T_{CPU} \le T_{delay} < T_{CPU} + T_{tlm}$ なる区間に一様に分布する。

これを $N_{FIFO} \neq 0$ のときにも拡張すると、T_{delay} は一般に、$T_{CPU} + N_{FIFO}T_{tlm} \le T_{delay} < T_{CPU} + (N_{FIFO} + 1)T_{tlm}$ なる区間に分布することがわかる。ただし、$N_{FIFO} \neq 0$ の場合には、T_{delay} の分布に

---

[1] FIFO には最大 4 イベントが入るので N_{FIFO} は 0 から 3 までの値をとる。

[2] Counting rate が極めて高い場合などには、「有効な」X 線検出が起こる時刻がテレメトリ編集時刻に依存することも考えられる。

は FIFO が詰まっている確率を考慮する必要があるため、その分布は上記の区間内で一様になるとは限らない。

まとめると、T_{arr}が一様な分布をする時には、

区間 $[T_{CPU}\quad, T_{CPU}+T_{tlm}\,]$ には、$N_{FIFO}=0$ のイベントが一様に分布
区間 $[T_{CPU}+T_{tlm}\quad, T_{CPU}+2T_{tlm}]$ には、$N_{FIFO}=1$ のイベントが分布
区間 $[T_{CPU}+2T_{tlm}\ , T_{CPU}+3T_{tlm}]$ には、$N_{FIFO}=2$ のイベントが分布
区間 $[T_{CPU}+3T_{tlm}\ , T_{CPU}+4T_{tlm}]$ には、$N_{FIFO}=3$ のイベントが分布

することがわかった。

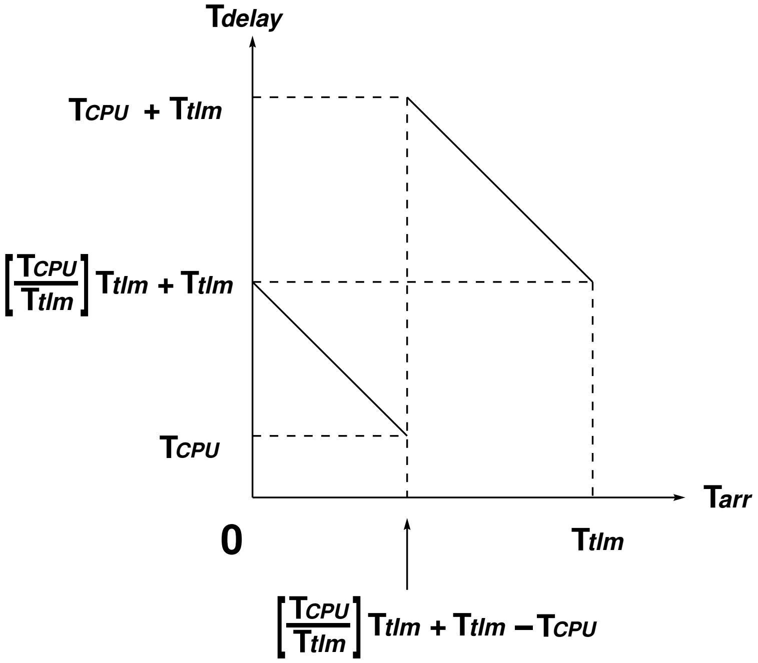

図 2: T_{delay}と T_{arr}の関係 ($N_{FIFO}=0$ のとき)。T_{tlm}と T_{CPU}が定数であることを仮定している。

3　実データを用いた解析

イベント時刻の定義が時刻ビットが "0" の時とそうでない時では異なるため、両者では本質的にイベント時刻の意味が異なり、その取り扱いには注意を要する。特に、時刻ビットが "0" の時には、イベント時刻 t_{tlm}は、CPU の処理時間や FIFO の混雑状況によっては、t_{arr}よりもかなり時間的に後になることが予想され、しかも、その遅れ具合いを推定することは困難である。そこで、その「遅れ具合い」を定量的に見積もるために、時刻ビットが十分にある観測で t_{tlm}と t_{arr}の両方を計算し、その差や、それぞれの時刻を用いた場合の folded light curve の違いなど、時系列解析に与える影響を調べてみた。

解析に用いたデータは、PV phase での PSR0540–69 の観測データで、bit-H のデータが全体の約 2/3 を占める。観測の詳細は表 1。解析は ASCA_ANL[3]上で GISREAD/GISTUNE[4]を用いて FRF から直接解析した。GIS では、ビットレートごとに時間分解能が異なるので、bit-H のデータのみを用いた。また、Elevation 5° 以上、COR 8 以上の時間のみを選択し[5]、さらに、マヌーバ直後の姿勢の不安定な時間も解析から除外した。

---

[3] ASCA_ANL version 0.99/0.99a。いずれもこのレポートを書いている時点では未公開。
[4] GISREAD は version 2.0/2.1、GISTUNE は version 0.7/0.8。いずれもこのレポートを書いている時点では未公開。
[5] timeregion version 1.3 (ASCA_ANL contrib)

観測日時	1993 年 9 月 23 日
FRF	ft930923_2248.2236
座標 (J2000)	$\alpha = 85.0460, \delta = -69.3326$
観測時間	$\sim$ 86ks (正味の露光時間は$\sim$ 40ks)
Counting rate	$\sim$ 2.4 c/s (GIS2), $\sim$ 2.8 c/s (GIS3)
GIS ビット配分	8-6-6-0-0-10

表 1: PSR0540–69 の観測の詳細情報

3.1 T_{delay}の分布

PSR0540–69 のデータを使って T_{delay}の度数分布を調べた。具体的には、fillcpudelay[6]を用いて、すべての観測データに対してt_{tlm}と t_{arr}の差をイベントごとに計算し、その度数分布をとった。イメージ上での切り出しや時刻補正などは一切していない。

結果を図 3に示す。図によると、約 8ms から 12ms にかけてほぼ一定の分布があり、この部分が全体のイベントの 95%程度を占めている。それよりも T_{delay}の短いイベントはほとんどない。また、全体のイベント数から見るとわずかではあるが、$T_{delay} > 12$ms の部分にもすそを引いている。また、T_{delay}の算術平均は、GIS2 で 10.6 ms、GIS3 で 10.7 ms である。

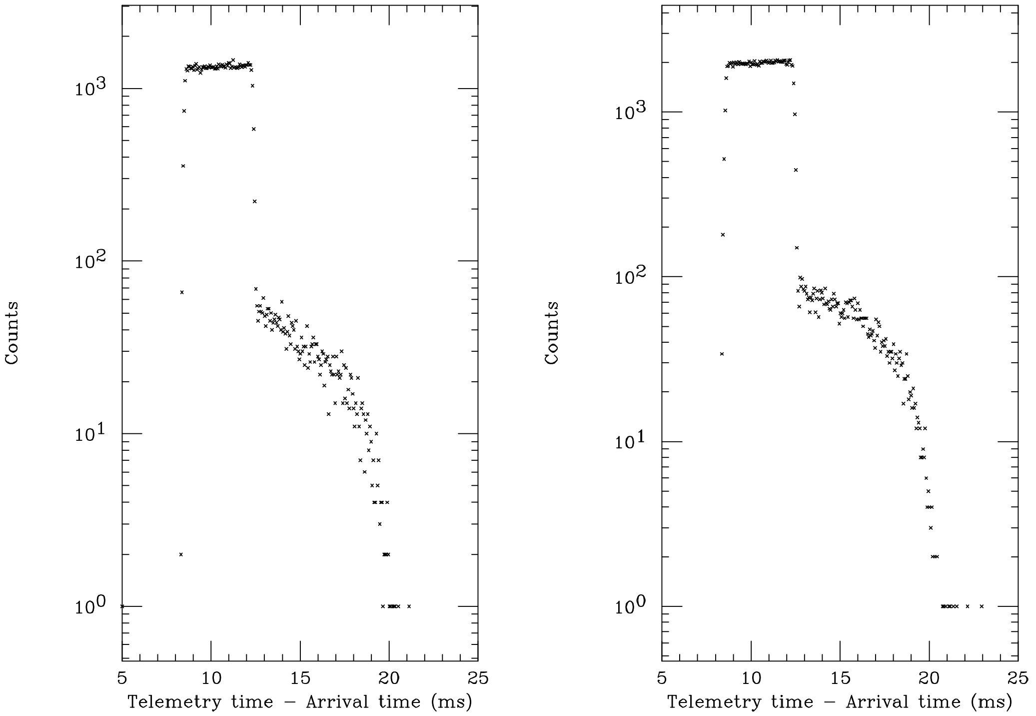

図 3: X 線が検出されてからそのイベントがテレメトリに出力されるまでにかかる時間 (単位は ms) の頻度分布。左図が GIS2、右図が GIS3 の結果。

この結果を§2の議論とつき合わせて考察してみる。まず、分布の下限値から、

$$T_{CPU} \simeq 8.5 \pm 0.1\text{ms}$$

であることがわかる。また、最も寄与の大きい部分の幅が約 4ms であることは、bit-H において、

$$T_{tlm} = 1/256\text{s} = 3.90625\text{ms}$$

---

[6] fillcpudelay version 1.0 (私家版)

であることと一致する。さらに、分布の上限値は§2によれば、

$$T_{CPU} + 4T_{tlm} \simeq 24.1\mathrm{ms}$$

となるはずであるが、図3のイベント分布はほぼこの範囲に収まっている。

3.2 Folding 解析

時刻付けの方法の違いが、実際の時系列解析にどのような影響を与えるかを調べるために、t_{tlm}を使った解析でPSR0540–69の周期を求め、その周期でのfolded light curveを作って、t_{arr}を使った解析で求めたものと比較してみた。この解析では、imageregin[7]を用いてソース近傍6分角のデータを切りだした。また、timeshift[8]を用いてGISの最小時間分解能以下の時刻を乱数で決定した。さらに、timecorrect[9]でbarycentric timeへの変換を行ない、pdotcancel[10]でPSR0540–69の周期の時間変化を打ち消す操作を行なった。Folding解析はchi2test[11]で行ない、イメージ切り出しの確認のため、gisimage[12]でX線像を同時にとった。

まず、periodgramを図4に示す。位相のビニングは50 bins/phaseにとったので、χ^2の自由度は49である。時刻ビットを使う場合とそうでない場合とでほぼ同様のperiodgramが得られたが、x=59.4付近のピークの有意度は、若干 時刻ビットを使った場合の方が高い。このperiodgramに対し、それぞれのピーク周辺を「Gaussian + 定数」なる関数でfitし、観測されたパルス周期を求めた。また、パルス周期のエラーは、fitting関数の値がピーク値からχ^2の1σ $(= \sqrt{2 * 49} \sim 9.9)$だけ下がった値よりも大きい範囲[13]とした。

こうして求まった周期は、

時刻ビットを使った場合	50.4184094 (14) ms
時刻ビットを使わない場合	50.4184094 (16) ms

で(括弧内は最終桁につくエラー)、時刻ビットを使う場合と使わない場合とで差は見られない。なお、図4中にも、求めた周期のベストフィット値を縦線で、周期の誤差を横向きのエラーバーでそれぞれ示した。

次に、求めた周期でのfolded light curveを図5に示す。図中、実線は時刻としてt_{arr}を使って得られるlight curve、破線は時刻としてt_{tlm}を使って得られるlight curveである。明らかに、破線が実線に対して約10ms遅れていることがわかる。このことは、t_{tlm}がt_{arr}に比べて平均的に10ms程度遅れていることを示している。

さらに、パルス位相の「ずれ」を定量的に見積もるために、t_{arr}を使って得られるlight curveとt_{tlm}を使って得られるlight curveとのcross correlation functionを計算した。得られたcross correlation functionを図6に示す。そして、これらのピーク付近を「Gaissian + 定数」でfitし、ピーク中心とそのエラー[14]を求めた。こうして求まったパルス位相のずれは、

GIS2のみ	10.5 ± 0.2 ms
GIS3のみ	10.7 ± 0.2 ms
GIS2+GIS3	10.6 ± 0.1 ms

---

[7] imageregion version 1.3 (ASCA_ANL contrib)

[8] timeshift version 1.0 (私家版)

[9] timecorrect version 1.1 (ASCA_ANL contrib)

[10] pdotcancel version 1.0 (ASCA_ANL contrib)

[11] chi2test version 2.0 (私家版)

[12] gisimage version 1.1 (私家版)

[13] ちなみに、こうして求まる周期の誤差範囲は、直観的には以下のように考えられる。まず、periodgramの各点にエラーとしてχ^2の1σのエラーバーをつける。そして、エラーバーの上端同士、下端同士をそれぞれつなぐと、その間にperiodgramに沿った細い領域ができる。データから得られたperiodgramを、この領域からはみ出さないように動かしてみる時、periodgramの中心が動き得る範囲が、だいたい本文中の方法で求まる周期範囲に一致する。

[14] ここではfittingで求まる普通のエラーを採用する

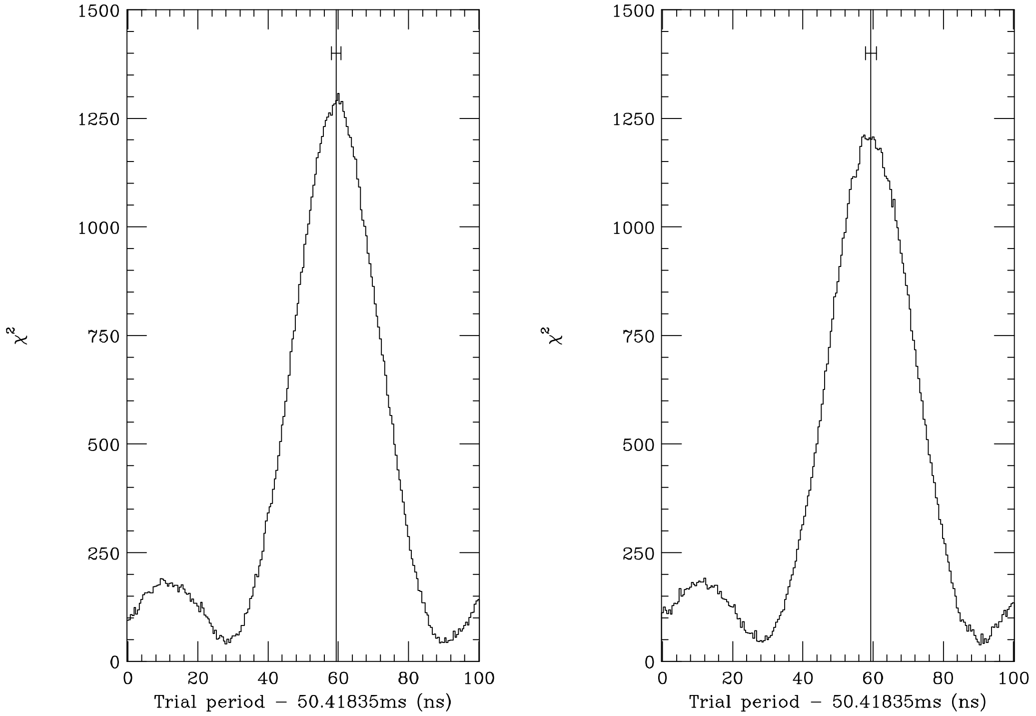

図 4: PSR0540–69 の periodgram (GIS2+GIS3)。位相のビニングは 50 bins/phase、したがって、χ^2の自由度は 49。左図が t_{arr}を使った解析、右図が t_{tlm}を使った解析の結果。いずれにも図中に周期のベストフィット値に縦線を入れ、周期の誤差を示すエラーバーをピークの上 (x=59.4, y=1400 の付近) に示した。

となり (エラーは 1σ)、互いに誤差の範囲で一致している。また、§3.1で求めた T_{delay}の平均値とも一致している。

最後に、求めた位相ずれを使ってパルス波形をずらして重ね、時刻ビットの有無によるパルス波形の違いを調べた。結果は図 6に示した通り、パルス波形に大きな違いは見られない。

4 まとめと今後の課題

本解析では、時刻ビット “0” の観測データでも 1/256s の時間分解能がでる場合があることがわかった。ただし、その場合、

- 時刻は約 10ms 遅れたものがついている。
  このため、パルサー解析などでのパルス位相などはずれる。周期を求める場合にはあまり問題がないようである。また、位相がずれて現れることを除けば、パルス波形もあまり影響されないようである。
- Counting rate が高い観測では信頼できない。
  この場合には、FIFO の占有率が高くなり、結果として、図 3の「すそ」の部分が相対的に大きくなる。そのため、T_{delay}の分布の広がり (標準偏差) が大きくなり、時刻の推定精度が下がることが予想される。

ということに注意する必要がある。また、本解析の副産物として、

$$T_{CPU} \simeq 8.5 \pm 0.1\text{ms}$$

であることもわかった。

また、時刻ビット “0” に関する今後の課題としては、

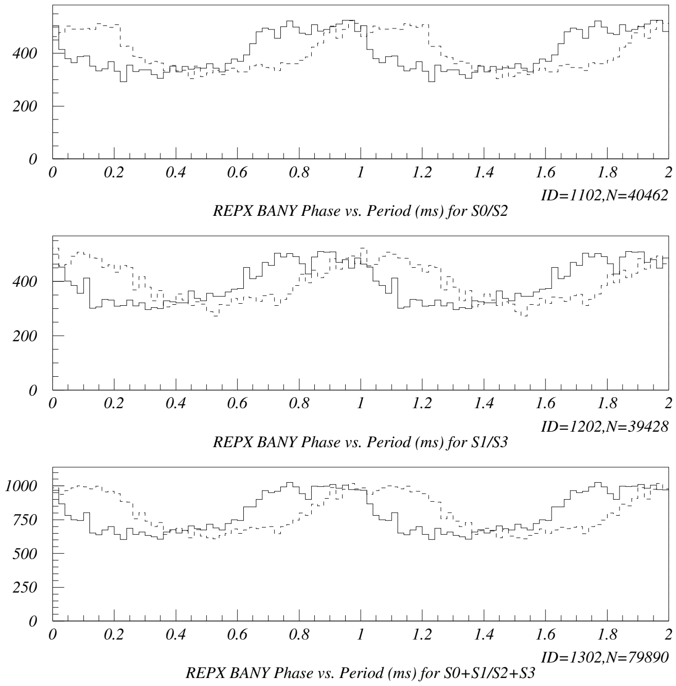

図 5: PSR0540–69 の folded light curve(2 周期分)。上段が GIS2 のイベントのみ、中段が GIS3 のイベントのみ、下段は GIS2+GIS3。各 light curve 中、実線は時刻として t_{arr} を使ったもの、破線は時刻として t_{tlm} を使ったもの。

- T_{delay} 分布の Counting rate 依存性
- Counting rate が高い場合の周期解析
- Counting rate の folded light curve への影響

などが考えられる。

5 謝辞

この解析を行なうにあたって、GISREAD/GISTUNE の改良を心良く引き受け、迅速に対応してくれた石崎君に感謝する。また、本レポートのテーマを調査するための方法を検討していただいた、長瀬先生および満田先生に感謝の意を表する。

参考文献

[1] 石崎 欣尚 修士論文 ASTRO-D 衛星搭載 GIS 信号処理系の較正 1993

[2] 科学衛星 ASTRO-D 中間報告書 1991

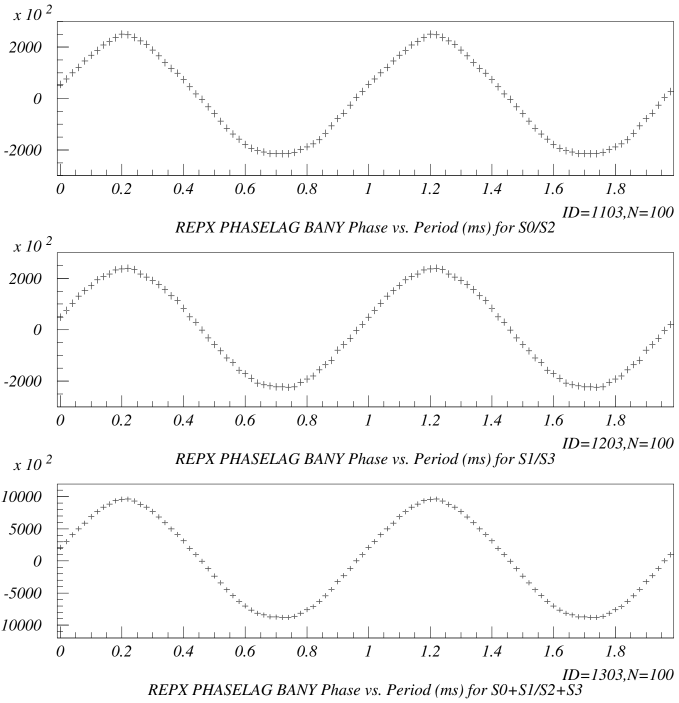

図 6: t_{arr}を使って得られる light curve とt_{tlm}を使って得られる light curve との Cross correlation function (2 周期分)。上段が GIS2 のイベントのみ、中段が GIS3 のイベントのみ、下段は GIS2+GIS3。

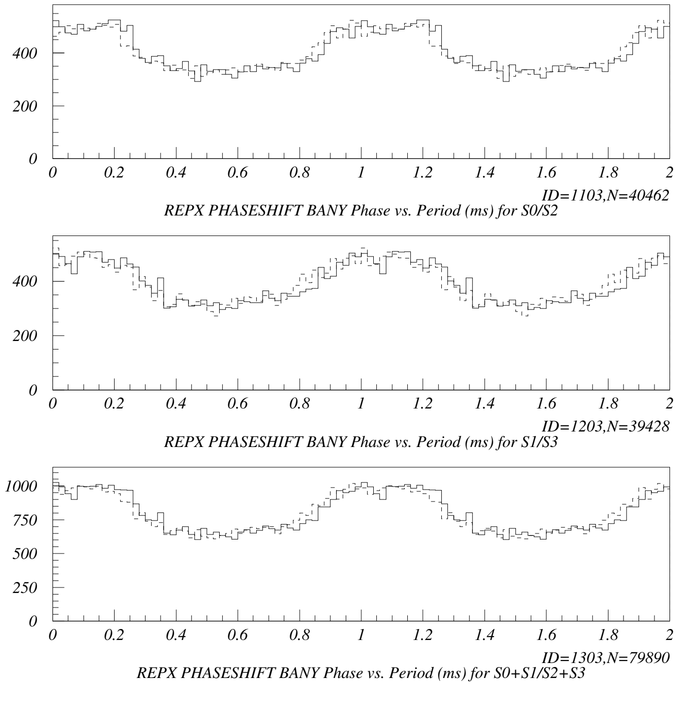

図 7: PSR0540–69 の folded light curve の形状の比較 (2 周期分)。パルス波形を比較するために、実線 (時刻として t_{arr}を使ったもの) を位相方向にずらしてある。それ以外は図 5と全く同じ。