

## III-3 レコーダ同期制御

雨谷 純\* 浜 真一\*

(昭和59年7月3日受理)

## 1. 概要

各 VLBI 局で取得されたデータが記録されている磁気テープをデータ・レコーダにかけ、相関の出ると予想される部分を同時に再生相関装置に出力するため再生する作業がレコーダ同期制御である。このためには、計算機から与えられる予測遅延時間差をもとに、270 インチ/秒という高速で走行している複数のデータ・レコーダを制御し、数百ビット (数百  $\mu\text{m}$ ) の精度で同期走行させる必要がある。同期に達するまでのデータは相関積分に関与しないので、できるだけ速やかに同期を達成しなくてはならない。本論文では主に、この制御方法と、制御に関するデータ・レコーダの機械的性能について報告する。

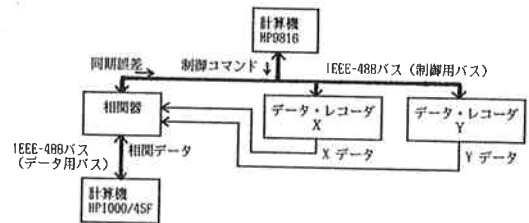
第1図は、K-3 データ・レコーダ (II-6 参照) の同期制御性能を調べるためデスクトップコンピュータで2台のデータ・レコーダの制御試験を行った際のセットアップである。実際の相関処理を行う時は、HP 9816 デスクトップコンピュータ (以下制御計算機と呼ぶ) が HP 1000/10L (データ収集制御計算機) に置き換わる。

制御は、IEEE-488 バスを介して行われる。制御計算機は、データ・レコーダに対し読み出しトラックの指定、テープ走行に関する命令、相関器に対し相関積分のおおまかな初期パラメータ設定命令を送出し、データ・レコーダから走行、停止状態のステータス、相関器から読み出しデータの時刻ラベル、同期誤差を受け取る (III-2 参照)。読み出しトラックや初期パラメータは、実際には HP 1000/45F (処理解析計算機) が観測時のログから読み取り、計算し、DS/1000 (II-12 参照) を通じて HP 1000/10L に受け渡すが、同期制御試験の際は、HP 1000/45F の演算結果を制御計算機のキーボードから入力した。

## 2. 制御方法

制御方法は、

- 1) ポジショニング; データの頭出し (データ・レコーダの立ち上げ加速分も考慮して巻き戻す)



第1図 同期制御性能テスト

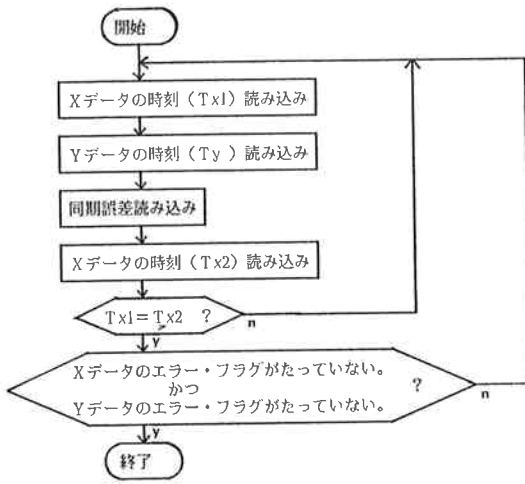
- 2) 粗制御; 先行しているレコーダの走行速度を下げる。(1/2の倍数)
- 3) 精制御; 先行しているレコーダの走行速度を下げる。(1/720ステップ)

の3つに分かれ、それぞれ同期誤差を 1)  $\pm 200$  キロビット 2)  $\pm 10$  キロビット 3)  $\pm 1$  キロビット以内に追込む事を目標としている。各制御はいずれも相関器から出力される同期誤差をもとに行われる。

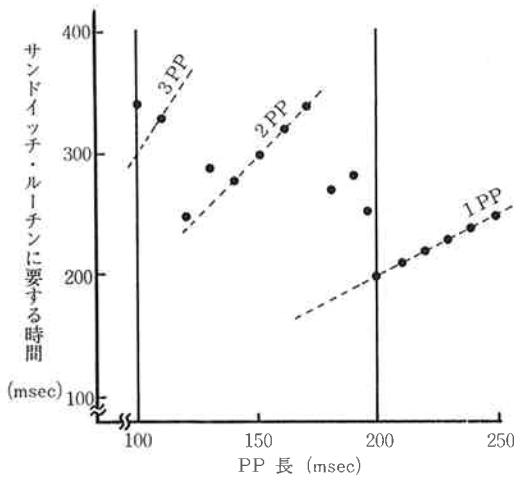
相関器は PP (Parameter Period; III-2 参照) ごとにテープ上の時刻を読み、同期誤差をビット単位で計算している。PP 長は相関器の制御用バスを介して制御計算機から 5 msec (1 フレーム) 単位で任意に設定できる。PP 長はできるだけ短い方が制御に要する時間 (すなわちデータ・ロス) が小さくなってよいが、相関器とのハンドシェイク時間により以下に述べるような制限が生じる。2.2, 2.3 で述べる各同期制御を行っている間は、ビット・スリップが多くほとんど時刻は読めないのので、時刻データがエラーでない PP の同期誤差を採用する必要がある。すなわち同期誤差データの前後に時刻データを採取し、前後の時刻が同じで、かつエラー・フラグのたっていない時に限り同期誤差情報を採用する様にする必要がある (これをサンドイッチ・ルーチンと呼ぶ。第2図参照)。

第3図は PP 長とサンドイッチ・ルーチンに要する時間の関係を示している。PP 長が 200 msec 以上の時は PP 長に比例してサンドイッチ・ルーチンにかかる時間も長くなる。しかし、PP 長を 200 msec より短くしてもサンドイッチ・ルーチンが 1 PP 分の時間内ではできなくなってくる。従って、PP 長を 200 msec に設定した時が一番効率よく同期誤差情報を採取できることがわ

\* 鹿島支所 第三宇宙通信研究室



第2図 サンドイッチ・ルーチン



サンドイッチ・ルーチンに要する時間  $T_s$  は、PP 長が 200 msec 以上の時は、PP 長を  $T_{pp}$  として  $T_s \approx T_{pp}$  であるが、PP 長がそれ以下になると、1 PP 分の時間ではサンドイッチ・ルーチンが終わらなくなってくる。PP 長が 140~170 msec の時は、サンドイッチ・ルーチンに要する時間が常に 2 PP を要するようになり、 $T_s \approx 2 \times T_{pp}$  となる。

第3図 サンドイッチ・ルーチンに要する時間

かる。もっともこれは制御計算機の演算速度に依存するから、HP 1000/10L で制御を行う際はまた異なる結果になる。今回の試験では PP を 200 msec とした。

### 2.1 ポジショニング

同期再生する際、データ・レコーダにテープをかけてまず最初にすべきことは、処理すべきデータの書かれている箇所を捜し出すことである。処理すべきデータを記録した時の状況(記録方向、フッター・カウント値等)は観測時にログに書かれているので、それをもとに容易

に捜し出すことができる。

処理すべき箇所を捜し出せても、そのまま 270 インチ/秒に立ち上げて処理を開始するわけにはいかない。テープの加速度は約 50 インチ/秒<sup>2</sup>なので、テープが 270 インチ/秒に立ち上がるまでに、テープは約 61 フィートも走行してしまう。加速途中はデータを読むことはできないので、この間のデータはむだになってしまう。そこでこのテープの立ち上げ分も見越してテープを巻き戻す必要がある(ポジショニング)。K-3 データレコーダには、フッター・カウント値を指定してやるとその指定値をめざして走行し、指定値の  $\pm 0.1$  フィートに入った所で自動的に停止する TPS (Tape Position) コマンドを設けたので、これを利用すれば容易にポジショニングすることができる。TPS コマンドで指定された目標値に達するとデータ・レコーダはサービスリクエストを発生し制御計算機にインターラプトをかけるようにしてある。

(1)式にポジショニング制御で戻すべきテープ長  $L$  を与える式を示す。TPS コマンドを使っても実際には、命令を受け取ってから動きだすまでの応答時間、立ち上げ加速の差等、2 台のデータ・レコーダに機械的な差があるので、式中の  $A_3$  項で各データ・レコーダについての補正を行っている。

$$L = s \times [(T_0 + A_1) - (T_r + A_2)] \times 135 + A_3 \quad \dots (1)$$

$L$ : 巻き戻すべきテープ長 [inch]

$s$ : フォワード時  $s = +1$   
リバース時  $s = -1$

$T_0$ : テープ上の時刻 [sec]

$T_r$ : 処理すべきデータの先頭時刻 [sec]

$A_1$ : 予測遅延時間 [sec] (基準となるデータ・レコーダでは  $A_1 = 0$ )

$A_2$ : 記録時に 135 inch/秒まで立ち上げる時間 [sec]

$A_3$ : 再生時、270 inch/秒まで立ち上げるのに要するテープ長 [inch]

現在ある 2 台のデータ・レコーダ ( $DR_x$ ,  $DR_y$ ) については

$$\begin{aligned} A_3 &= 1162 \text{ inch for } DR_x \\ &= 1052 \text{ inch for } DR_y \end{aligned}$$

まず低速で再生し、その時のテープ上の時刻とフッター・カウント値を記憶する。それをもとに(1)式より  $L$  を計算し、現在のフッター・カウント値に加えたものを TPS コマンドでおくってやればよい。

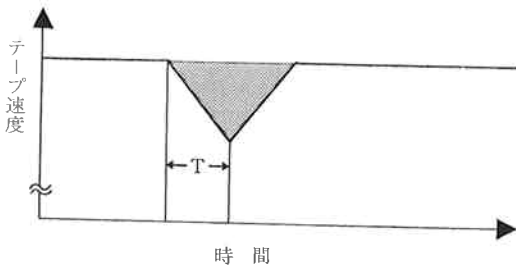
この制御方法を用いることにより  $\pm 200$  キロビットの精度でポジショニングを行うことができる。もし制御後の同期誤差が 200 キロビットを超えてしまった場合は、

再度ポジショニングをおこなう事にしている。これは次の粗制御によるデータの損失を極力少なくするためである。ポジショニングを繰り返し行うことは、実際の処理時間を浪費するが、データの損失にはつながらない。

2.2 粗制御 (SPD 制御)

ポジショニングにより同期誤差が200キロビット以下となつて、270インチ/秒で走行している2台のデータ・レコーダに、さらに制御をおこなつて同期誤差を縮めるのが粗制御である。

粗制御は先行しているデータ・レコーダの走行速度を一定時間下げて同期誤差を吸収しようとするもので、第4図に示すような速度変化を利用する。第4図の斜線部の面積が制御により変化した同期誤差となる。K-3データ・レコーダの母体となっているM-96データ・レコーダはランプ・ジェネレータ回路(第5図)を持っていて、走行開始時や、走行中に速度変更命令 (SPD コマンド) を受けた時の加速度がほぼ一定である。従つて270インチ/秒で走行している時に、走行速度を135インチ/秒に変更する命令を送り、一定時間の後にまた走行速度を270インチ/秒に戻してやると、時間軸を見た速度の変化は台形もしくは三角形になる。台形の制御になるのは同期誤差が約11メガビット以上になった時であ



第4図 粗制御における速度変化

る。しかしながら、既にポジショニングにより同期誤差が300キロビット以下になっているので、速度変化としては第4図に示す三角形になるような制御しか行わない。

第6図に粗制御の制御時間と、制御により変化したビット数の関係を示す。速度変化の様子(第4図)から2次式で近似されることが予想される。

$$T = A_1 + A_2 \times \sqrt{|B|} + A_3 \quad \dots\dots(2)$$

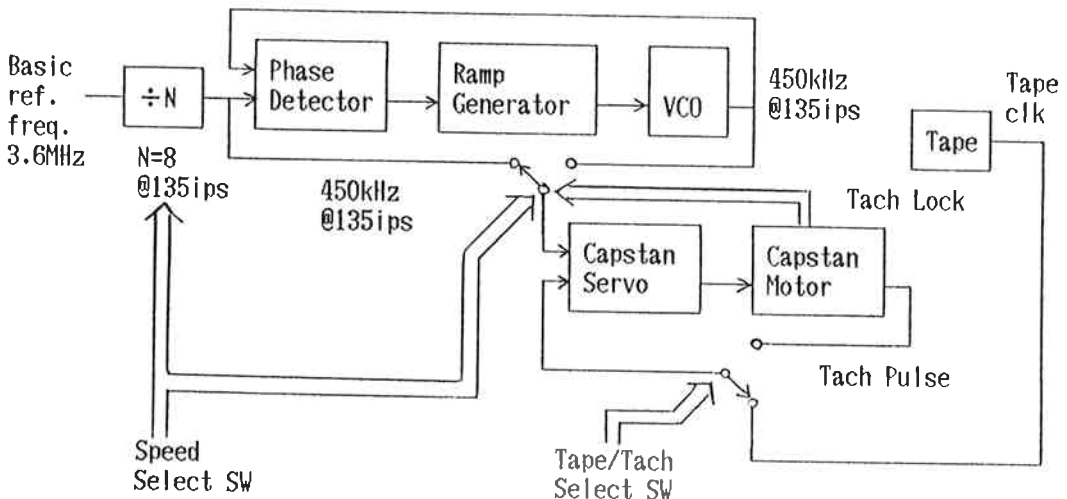
T : 制御時間 [msec]

B : 同期誤差 [bit]

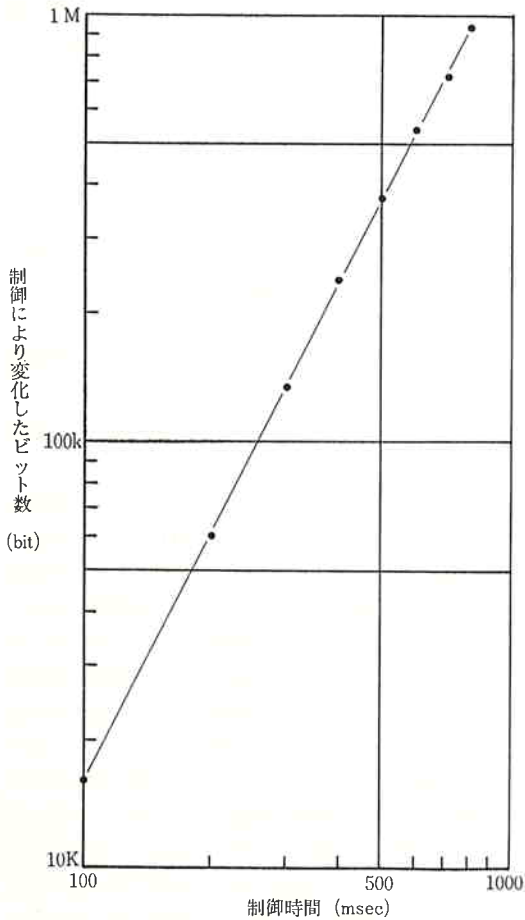
もし速度変化が完全な三角形で、データ・レコーダに機械的差がないとするなら  $A_1, A_3$  はともに0、 $A_2$  はデータ・レコーダの加速度を50インチ/秒<sup>2</sup>として0.7740となる。

粗制御を行うには、先行するデータ・レコーダに対し走行速度を135インチ/秒にするSPDコマンドを出した後、(2)式で計算したT[msec]だけ待って走行速度を270インチ/秒にするSPDコマンドを送ればよい。

さて(2)式で計算される制御時間は、あくまでSPDコマンドを送る間隔を決定するものであり、実際に制御を開始してから終了するまでの時間は2×Tよりずっと長くかかる。これは速度の急激な変化によりテープ上のデータが一時読めなくなってしまうため、走行が安定するまで相関器からの同期誤差情報が得られず次の制御に移れないからである。制御されるビット数とその制御に要する時間(データ上の時間)の関係を第7図に示す。制御されるビット数が小さくなると、制御に要する時間は約2.1秒に漸近する。すなわち粗制御で同期誤差を吸収しようとした場合、必ず2.1秒以上のデータ・ロスが生じてしまう。一方、次に述べる精制御の方は、この制御に要する時間が、制御すべきビット数の少ない時に粗



第5図 コントロール・ロジック回路



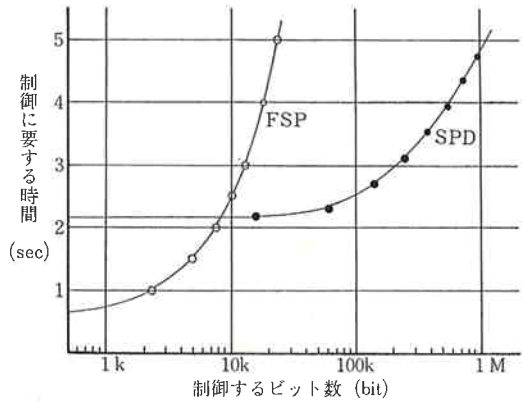
第6図 粗制御により制御されるビット数

制御に比べずっと小さくなる。およそ10キロビットを境に精制御の方が制御に要する時間が短くなるのがわかる。

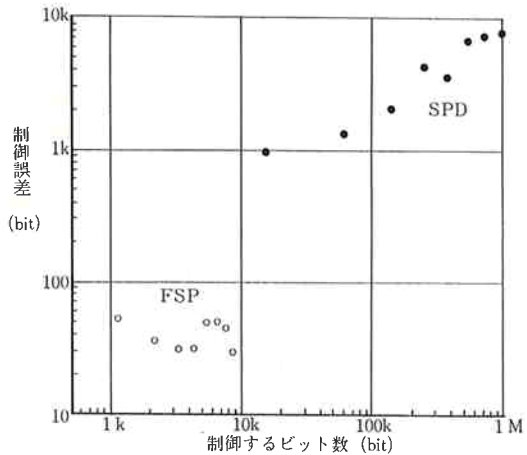
第8図に粗制御と精制御における、制御すべきビット数と制御誤差の関係を示す。粗制御10キロビット～200キロビットの同期誤差を補正しようとした場合、数キロビットの制御誤差を伴う。これに対し精制御は10キロビット以下の同期誤差を補正しようとする限り、常に100ビット以下の制御誤差に納まっている。

以上のことから粗制御は同期誤差が10キロビット～200キロビットの範囲にきた時に行うこととした。第6図に対しこの範囲で最小二乗法を適用し(2)式の係数を決定すると、

$$\begin{aligned}
 A_1 &= -1.196 \text{ for } DR_x \\
 &= -8.267 \text{ for } DR_y \\
 A_2 &= 0.8550 \text{ for } DR_x \\
 &= 0.8361 \text{ for } DR_y \\
 A_3 &= 602.2 \text{ for } DR_x
 \end{aligned}$$



第7図 制御に要する時間



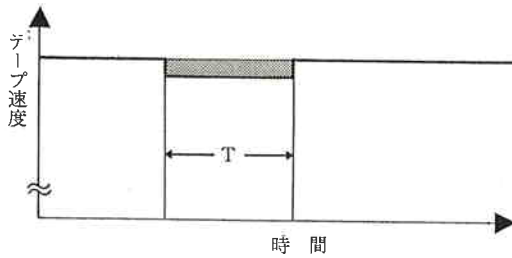
第8図 制御誤差

$$= 983.3 \text{ for } DR_y$$

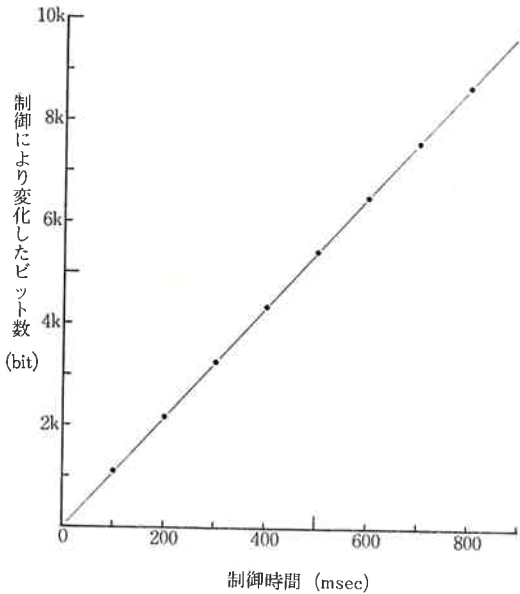
となり、残差は最悪でも1キロビット程度となった。

### 2.3 精制御 (FSP 制御)

以上、述べてきたように SPD コマンドによる粗制御では同期誤差を確実に1キロビット以下に納めることはできない。K-3 データ・レコーダはキャプスタン・モータ回転の基準周波数を微小に変化させることのできるシンセサイザーを装備しており、これを使うことにより、テープ走行速度を微小 (270.000 インチ/秒から 269.625 インチ/秒) に変化し、同期誤差を1キロビット以下に追込むことが可能である。K-3 データ・レコーダの基準周波数は通常 3.6 MHz であるが、上記のシンセサイザーを使うことにより 5 kHz ステップで変化させることができる。第9図は基準周波数を変化させた時のデータ・レコーダの速度変化を模式的に描いたもので、ほぼ矩形に変化することが予想される。この制御は FSP というコマンドを用い制御計算機から行うことができる。



第9図 精制御における速度変化



第10図 精制御により制御されるビット数

第10図にこの制御の制御時間と、制御により変化させたビット数の関係を示す。FSP コマンドを用い基準周波数を5kHz だけ下げ一定時間待った後に3.6MHz に戻してやった時の同期誤差の変化を表している。第10図に示されるように直線性が非常によく一次式でフィッティングすると、残差は最大の所でも10ビットに満たない。また、2.2 で述べた様に、同期誤差が10キロビット以下の範囲に適用するならば制御に要する時間が短く(第7図)、制御誤差も小さい(第8図)。式(3)に第10図より求めた精制御の制御時間を求める式を示す。

$$T = A_1 \times |B| + A_2 \quad \dots(3)$$

$T$  : 制御時間 [msec]

$B$  : 同期誤差 [bit]

$A_1 = 0.09268$  for  $DR_x$

$= 0.09274$  for  $DR_y$

$A_2 = 2.9$  for  $DR_x$

$= 1.3$  for  $DR_y$

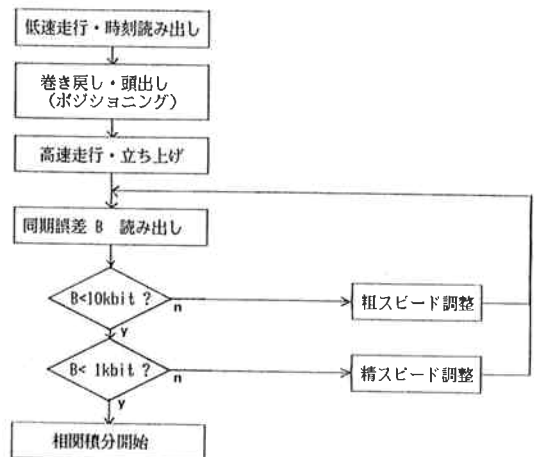
第9図のような理想的な矩形形状に速度が変化しているとするならば、1ステップの速度変化が0.375インチ/秒ゆえ  $A_2 = 0$ ,  $A_1 = 0.08$  となる。

FSP 制御が安定している理由は、速度の変化量が小さいこと、テープ・モードで走行している事による。テープ・モードは、テープ上のデータ・クロックでデータ・レコーダのキャプスタン・サーボ回路にフィード・バックをかけるもので、通常走行(タック・モード)に比べ格段にビット・レートが安定する。2台のデータ・レコーダをタック・モードで走行させた場合、同期誤差の変化は毎秒数100ビットにも及ぶが、テープ・モードでは一分間に10ビット程度である。ただしテープ・モードでは、データ・クロックの中断や、テープ・スリップの時にキャプスタンが異常回転を起こしバキューム異常でレコーダが停止してしまう。このためにSPD制御のように速度変化が大きく、データの読み出しが不能になるような時は、テープ・モードは使用できない。そのためポジショニングもしくは粗制御で同期誤差が10キロビット以下になってから、タック・モードをテープ・モードに切り換えるようにしている。第5図にデータ・レコーダのコントロール・ロジックを示した。

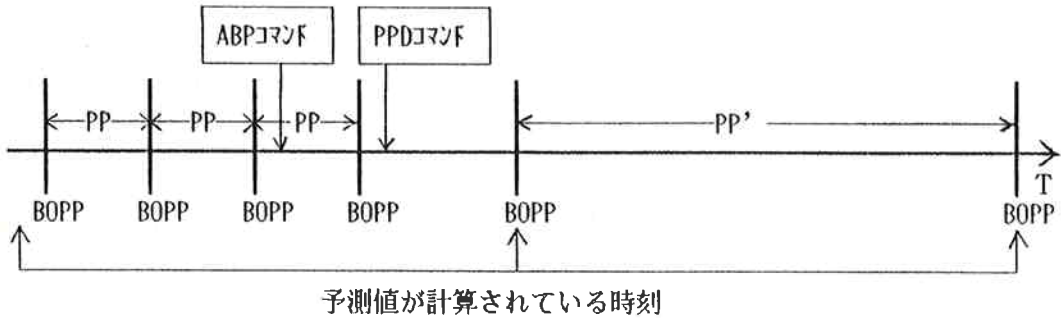
### 3. 同期制御とデータ・ロス

第11図に以上述べてきた各制御を行うフローチャートを示す。同期誤差が1キロビット以下になり、テープ・モードで安定走行するようになってレコーダの同期制御は一応終了する。しかし実際の相関処理では、さらにPP長を相関積分に適した長さに変更し、また各PPの先頭時刻が予測値を計算した時刻にあうようにそろえる必要があり、この制御は相関器に対して行われる。

相関器には、PP長を次に来るPPに限り変えることのできるABPコマンドがある。このコマンドを使い、第12図に示すタイミングで次式により与えられるフレーム数を送る。

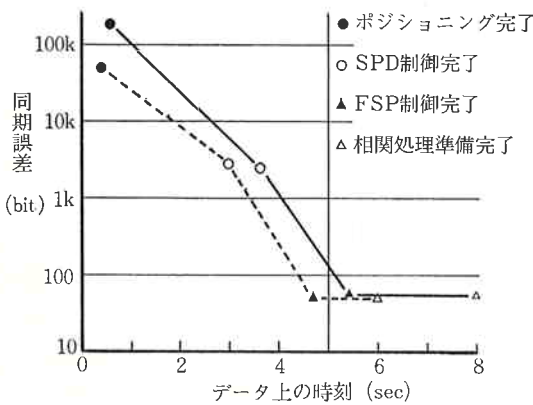


第11図 同期制御フローチャート



PP ;制御用PP長      PP' ;相関積分用PP長      BOPP ;Begin of Parameter Period

第12図 コマンド送出タイミング



各同期制御を行った際の同期誤差の推移を2例(実線と点線)示す。横軸は記録開始時刻を0とした時の再生データの時刻である。

第13図 同期制御による同期誤差の変化

$$N_{abp} = \text{MOD}((T_{corr} - T_v - 3 \times T_{pp1}), T_{pp2}) / 0.005 \quad \dots(4)$$

- $N_{abp}$  : 増加させるフレーム数
- $T_{corr}$  : 予測値を計算してある時刻 [sec]
- $T_v$  : テープ上の時刻 [sec]
- $T_{pp1}$  : 同期制御中に使用したPP長 [sec]
- $T_{pp2}$  : 相関積分を行うPP長 [sec]
- $\text{MOD}(a, b)$  ;  $a$  を  $b$  で割った余り。

そして、その次のPPを待って、PPDコマンドを使いPP長を実際の相関処理を行うのに適した長さに変更すればよい。

以上の制御を行いPPがそろったのを確認した後、SPRコマンドで相関器の制御権をHP 1000/45Fに移し、ESRコマンドでPPごとにHP 1000/45FにSRQインターラプトがかかるようにして相関処理のための同期制御のすべてが完了する。第13図に相関処理を行うPP長を2秒とした時の、一連の制御の間の同期誤差の変化を示す。すべての制御が完了するまでに失われるデータの量は10秒(40メガビット)以下になることがわかる。

VLBIの一般的観測時間は一つの星について約180秒である。従って同期制御により失われるデータ量は、全体の約6%にすぎない。

#### 4. 今後の課題

##### 4.1 同期制御ソフトウェア

今回、各制御の実験式を求めたが、機械的部分の多いデータ・レコーダの制御であり、経年変化が予想される。同期制御ソフトウェアは、過去の一定期間の同期制御の履歴を記憶し、それをもとに実験式を更新してゆくような学習効果のあるものに変更することがのぞましい。

##### 4.2 データ・レコーダ

現在のK-3データ・レコーダはデータの誤り訂正を行っていないので、もし、PPの先頭のフレームで時刻符号が読み取れなかった場合PP単位でデータが欠損してしまう。これは制御時間の浪費につながるし、また、BOPP(PPの変わり目)が検出できないために3.で述べたPP操作を失敗してしまう。時刻符号のみの誤り訂正はぜひ追加したい。

また、できればさらにデータ・レコーダのインテリジェンスを拡張し、制御計算機の負担を軽減してゆきたい。すなわち現在、制御用とデータ収集用の2台の計算機を使用しているが、制御用計算機の機能をすべてデータ・レコーダに持たせて、一台の計算機だけで相関処理が行えるようにしたい。これは、3台以上のデータ・レコーダの同期再生を同時に行うマルチ・ベースライン処理をする際に特に威力を発揮すると思われる。

##### 4.3 相関器

現在の相関器の出力する同期誤差情報にはエラー・フラグがなく、また、同期制御を行う計算機にSRQインターラプトがかからない。このため先に述べたサンディッチ・ルーチンを行う必要が生じ制御計算機の負担は増し、制御時間も短縮できない。こうした点も考えた相関器の改良が望まれる。

### 5. ま と め

K-3 データ・レコーダを使用し, 迅速かつデータ・ロスの少ない同期制御方法を開発した. Mark-III システムで行われている同期制御はテープ速度が我々の半分の135インチ/秒でありデータ・ロスはほぼ同じである. これまで相関処理を一手に引き受け行ってきた米国

のヘイスタック観測所では, 相関処理が追いつかず未処理テープが山積しているということなので, 本研究の同期制御方法を用い高速処理を行うことは, 今後の膨大なVLBI データ処理に大いに役立つものと思われる.

最後に, 日ごろ御指導頂く第三宇宙通信研究室の河野室長, また, 本研究に御協力下さった杉本研究官, 国森技官, 金子技官に深く感謝いたします.

