

Consiglio Nazionale delle Ricerche

VSAPL:

Un sistema grafico interattivo per l'analisi dei dati

telemetrici relativo all'assetto del satellite Sirio.

preparato da:

G. FACONTI

P. PALAMIDESE

142

CNUCE

Istituto del Consiglio Nazionale delle Ricerche

Pisa, Aprile 1978

Copyright - Aprile 1978

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

VSAPL:

Un sistema grafico interattivo per l'analisi dei dati telemetrici relativi all'assetto del satellite Sirio.

preparato da:

G.FACONTI

P.PALAMIDESE

**CNUCE - Istituto del Consiglio Nazionale delle Ricerche.
Pisa, Aprile 1978**

Indice

1. Introduzione

2. Generalita'

- 2.1 Accesso al sistema
- 2.2 Implementazione APL
- 2.3 Unità di Input/Output
- 2.4 Lo spazio di lavoro
- 2.5 Lista dei comandi

3. Uso del TPS

- 3.1 Istruzioni
- 3.2 Descrizione dei comandi
- 3.3 Uso del sistema
- 3.4 Messaggi di errore

4. Bibliografia

APPENDICE A - Listings delle funzioni.

APPENDICE B - Cross reference dello spazio di lavoro.

APPENDICE C - Esempio di seduta al terminale.

1. Introduzione

Nel presente lavoro vengono descritte la struttura e le modalita' di utilizzo del Telemetry Preprocessor Subsystem (TPS) che e' parte del Flight Dynamic System (FDS) utilizzato per il controllo del volo del satellite SIRIO.

Il compito del TPS e' quello di permettere all'utente di compiere un'analisi qualitativa dei dati di telemetria trasmessi dal satellite ed acquisiti dalla stazione di terra del Fucino. I dati di telemetria, dopo una prima elaborazione sul PDP 11/70 del Fucino, sono trasmessi in real-time su di una linea a 1200 bps agli elaboratori del CNUCE dove il programma SIRLINK, che controlla tale trasmissione, provvede a memorizzarli su disco.

I dati cosi' ottenuti sono utilizzati essenzialmente per la determinazione dell'inclinazione dell'asse di spin del satellite rispetto al suo piano orbitale e come tali vengono dati in input all'Attitude Determination Subsystem dell'FDS.

Poiche' la determinazione dell'assetto del satellite e' effettuata sulla base di analisi statistiche dei dati di telemetria distribuiti su una intera orbita, secondo una funzione del modello di soluzione adottato, il TPS fornisce un valido aiuto all'utente nella scelta dei blocchi di dati da selezionare.

Infatti la mole dei dati di cui si tratta (5400 frames di 1040 bytes in un'orbita) renderebbe arduo il compito di selezionare gli intervalli di dati validi qualora questo fosse effettuato esclusivamente su basi numeriche.

La caratteristica fondamentale del TPS e' infatti quella di essere un programma interattivo che utilizza le possibilita' grafiche del Tektronix Display Terminal 4015. Questa caratteristica offre all'utente la possibilita' di visualizzare immediatamente l'andamento qualitativo di tutte le grandezze che entrano in gioco nel calcolo dell'assetto e di ridurre quindi l'analisi numerica a quegli intervalli di tempo in cui i dati hanno un andamento 'ragionevole'.

L'implementazione del TPS e' stata effettuata utilizzando il linguaggio APL, che e' dimostrato essere il mezzo piu' conveniente nell'implementare sistemi conversazionali sia dal punto di vista della programmazione che dell'utilizzo.

2. Generalita'

Il TPS e' un package implementato in linguaggio APL sotto VM/370. Nei seguenti paragrafi e' data una descrizione generale sia del package che dell'ambiente in cui opera.

2.1 Accesso al sistema

Il sistema TPS e' costituito da uno spazio di lavoro APL che puo' essere utilizzato seguendo la procedura descritta di seguito.

Una volta stabilita la connessione con l'elaboratore secondo le modalita' richieste dal terminale usato, l'utente stabilisce il collegamento con il VM/370 dando il comando:

```
LOGON <userid>
```

dove <userid> e' il nome della macchina virtuale dell'utente. Dopo che il VM/370 ha richiesto la password della macchina virtuale e questa e' stata introdotta correttamente, l'utente si trova in ambiente CP. Ogni legittimo comando di CP puo' essere dato in quest'ambiente. Uno dei possibili comandi e':

```
DEFINE STOR 1M
```

Questo comando definisce la dimensione della memoria della macchina virtuale per una corretta esecuzione di TPS. Un altro comando che puo' essere dato e':

```
LINK <sid> <vaddr1> <vaddr2> RR <password>.
```

Questo comando rende disponibile per la macchina virtuale dell'utente il minidisco di indirizzo virtuale <vaddr1> della macchina virtuale <sid> in modo read/only assegnandogli l'indirizzo <vaddr2>; <password> e' la parola chiave che autorizza l'operazione. Il disco che e' reso disponibile deve essere quello contenente i dati di telemetria da elaborare.

IPL e' un altro comando di CP che e' usato per inizializzare un sistema operativo. Col comando:

```
IPL CMS
```

il sistema operativo CMS e' caricato nella macchina virtuale dell'utente che puo' cosi' dare qualsiasi comando di CMS ed in particolare il comando:

```
ACCESS <vaddr2> B.
```

Con tale comando l'utente notifica al CMS l'esistenza del minidisco di indirizzo <vaddr2>, definito dal comando LINK precedente, che viene acceduto come B-disk.

A questo punto l'utente puo' entrare in ambiente APL col comando:

```
VSAPL APL100 APL101 APL110
```


Le opzioni che seguono il comando VSAPL sono i nomi degli auxiliary processors utilizzati durante l'esecuzione del TPS.

Una volta che il VSAPL e gli auxiliary processors sono stati caricati puo' essere dato il comando APL:

)LOAD <libnum> TPS

dove <libnum> e' il numero della libreria pubblica del VSAPL contenente lo spazio di lavoro TPS. Con questo comando vengono caricate nello spazio di lavoro attivo tutte le funzioni del sistema TPS che puo' essere cosi' attivato tramite il comando:

START

che esegue la funzione dello stesso nome. Questa funzione attiva l'ambiente conversazionale e da questo momento l'utente puo' mandare tutti i comandi di TPS che sono dettagliatamente descritti nel seguito. Tutti i comandi che seguono il comando:

IPL CMS

possono essere attivati dal solo comando:

TPS>

2.2 Implementazione APL

Il sistema TPS e' costituito da un insieme di funzioni contenute nello spazio di lavoro TPS. Queste funzioni sono state implementate sotto il sistema VSAPL.

Le grandezze caratteristiche dello spazio di lavoro sono poste ai valori di default del VSAPL tranne che per la dimensione dell'area di memoria utilizzata che deve essere definita ad 1M come specificato nel paragrafo 2.1.

Il sistema VSAPL deve inoltre essere predisposto all'utilizzo degli auxiliary processors di cui al paragrafo 2.1. Gli auxiliary processors sono dei programmi che forniscono servizi al VSAPL tramite il meccanismo delle shared variables permettendo una completa interazione tra l'APL e il sistema che lo ospita.

Una shared variable e' associata ad una zona di memoria a cui possono accedere due processors. Se un processor vuole comunicare all'altro una informazione la scrive nella variabile e l'altro processor, successivamente, acquisisce l'informazione leggendola dalla stessa variabile. Ad un dato istante percio' la variabile ha soltanto un valore: l'ultimo valore ad essa assegnato da uno dei due interlocutori. Sintatticamente una shared variable e' indistinguibile dalle altre variabili APL e puo' essere usata normalmente alla destra o alla sinistra dell'operatore di assegnazione.

Tra gli auxiliary processors disponibili sotto VSAPL quelli usati dal TPS sono i seguenti:

APL100 viene usato per inviare comandi di CP e di CMS
APL101 viene usato per immagazzinare in un'area di memoria dei dati che verranno utilizzati alla successiva richiesta di input
APL110 viene usato per leggere o scrivere files su disco.

2.3 Unita' di input/output

Il sistema TPS necessita come supporto di I/O di un Tektronix Display Terminal 4015, terminale interattivo fornito di un video a memoria e di una tastiera alfanumerica. Questo terminale puo' lavorare sia in modo alfanumerico che in modo grafico passando alternativamente da uno stato all'altro a seconda delle necessita' espresse dal programma.

A questo terminale puo' essere collegata una hard-copy che puo' riprodurre su carta le immagini che compaiono sullo schermo ad un certo istante.

2.4 Lo spazio di lavoro

Diamo di seguito una descrizione delle principali variabili e funzioni che sono contenute nello spazio di lavoro TPS.

FRAME e' una variabile definita come una matrice numerica di rango 2 con tante righe quante sono le frames di telemetria caricate nello spazio di lavoro ed 80

colonne. Ogni riga rappresenta una major frame contenente 16 minor frames ciascuna delle quali e' costituita dai seguenti 5 campi:

1. tempo della mjrnor frame in millisecondi di anno
2. flag che indica la bonta' dei dati trasmessi
3. numero che identifica la frame entro la major frame
4. primo dei due intervalli di tempo trasmessi nella mjrnor frame in termini di clock-counts
5. secondo dei due intervalli di tempo trasmessi nella mjrnor frame in termini di clock-counts.

Gli ultimi due campi contengono le misure rilevate dai sensori a bordo del satellite come descritto nella seguente tabella. In prima colonna e' indicato il numero d'ordine della mjrnor frame all'interno di una major frame, nella successiva colonna la sorgente della misura e nell'ultima colonna il suo significato.

| | | | |
|----|--------------|----------|-------------------------------------|
| 0 | INF1 A | INF2 A | EARTH WIDTH |
| 1 | INF1 B | INF2 B | EARTH WIDTH |
| 2 | SVB B | SVB A | SUN ANGLE |
| 3 | ACL | ACL | |
| 4 | ILF B | ILF B | SUN TO EARTH-IN SUN TO EARTH-OUT |
| 5 | ILF A | ILF A | SUN TO EARTH-IN SUN TO EARTH-OUT |
| 6 | SPF A | SPF B | SPIN RATE |
| 7 | ACL | ACL | |
| 8 | INF1 B | INF2 B | EARTH WIDTH |
| 9 | INF1 A | INF2 A | EARTH WIDTH |
| 10 | SUN SIGHTING | EARTH-IN | |
| 11 | SVB A | SVB B | SUN ANGLE |
| 12 | ILF A | ILF A | SUN TO EARTH-IN SUN TO EARTH-OUT |
| 13 | ILF B | ILF B | SUN TO EARTH-IN SUN TO EARTH-OUT |
| 14 | ACL | ACL | |
| 15 | SPF B | SPF A | SPIN RATE |

TABCOM e' una variabile definita come una matrice di caratteri di rango 2 costituita da tante righe quanti sono i comandi del TPS; ogni riga contiene il nome di un comando. Questa matrice e' utilizzata dalla funzione START per verificare la correttezza dei comandi inviati dall'utente.

FR e' la variabile messa in sharing col processor 110. Alla variabile FR viene assegnato il seguente valore iniziale:

'TLMTRY FILE B (BIT'

dove 'TLMTRY FILE B' identifica il file di CMS che contiene i dati di telemetria e l'opzione BIT indica che i dati sono codificati in binario puro. Ogni

successivo riferimento alla variabile FR produce la lettura di un record di telemetria dal file su disco o di un blocco di records a seconda delle specifiche contenute nella variabile di controllo FC. Dopo ogni lettura il contenuto di FR e' convertito da binario a decimale e concatenato alla matrice FRAME.

FC e' la variabile di controllo messa in sharing col processor 110. Viene inizializzata al seguente valore:

'TLMTFY FILE B (CTL'

dove 'TLMTFY FILE B' identifica il file di telemetria e l'opzione CTL dice che si tratta della variabile di controllo. Ogni successivo riferimento alla variabile FC produce come risultato un vettore numerico a quattro componenti aventi il seguente significato:

1. codice di ritorno
2. puntatore al record del file su disco da leggere
3. puntatore al record del file su disco da scrivere
4. numero di records che vengono letti o scritti con una sola operazione

Tramite la variabile di controllo il sistema puo' gestire un accesso non sequenziale al file di telemetria.

SV100 e' la variabile messa in sharing con il processor 100. Tale variabile permette di dare comandi di CMS e di CP dall'APL. A questa variabile non viene attribuito alcun valore iniziale. Il sistema gestisce la variabile su richiesta dell'utente il quale deve specificare con quale ambiente vuole comunicare.

SV101 e' la variabile messa in sharing col processor 101. Ad essa e' assegnato il valore iniziale:

'CMS (APL BEG'

In questo modo viene definita un'area di memoria nella quale si possono registrare vettori di caratteri. La prima opzione APL indica il tipo di rappresentazione riservata ai caratteri scritti nell'area, la seconda opzione BEG indica che l'area e' organizzata come una struttura a pila (LIFO) e come tale verra' svuotata alla prima richiesta di input. Piu' precisamente questa variabile viene utilizzata per eseguire una successione di comandi di APL e di CMS.

PLOT e' la funzione che esegue la richiesta dell'utente di disegnare dei grafici da quei dati di telemetria che sono specificati di volta in volta. I grafici sono realizzati utilizzando le possibilita' offerte

dal terminale video Tektronix 4015. Tramite caratteri di controllo e' possibile interagire con il terminale e controllarne alcune operazioni. In particolare la variabile GS contiene dei caratteri di controllo che fanno passare il terminale dallo stato alfanumerico a quello grafico. In questo ambiente e' possibile indirizzare il beam in un punto qualsiasi dello schermo. Successivi indirizzamenti dello schermo vengono considerati come punti da collegare. Infine la variabile US contiene dei caratteri di controllo che fanno tornare di nuovo il terminale in ambiente alfanumerico. Prima di disegnare un grafico il sistema provvede a cancellare lo schermo del terminale automaticamente.

2.5 Lista dei comandi

Tutti i comandi di TPS possono essere raggruppati, a seconda delle operazioni che eseguono, nel seguente modo:

1. comandi di inizializzazione

START : questo comando connette l'utente con l'ambiente conversazionale del TPS. Da questo momento l'utente puo' inviare tutti i comandi di TPS.

2. comandi per interagire con altri sistemi

CP : questo comando deve essere usato quando si vuole inviare comandi di CP senza uscire dal TPS.

CMS : questo comando deve essere usato quando si vuole inviare comandi di CMS senza uscire dal TPS.

3. comandi per avere informazioni sul sistema

COMDS : questo comando serve per avere la lista di tutti i comandi e la loro sintassi.

HELP : serve per avere la sintassi di un comando alla volta.

STATUS : questo comando serve per avere informazioni sui dati contenuti nello spazio di lavoro attivo.

4. comandi per la manipolazione dello spazio di lavoro

SHAPE : questo comando serve per inizializzare le shared variables FR ed FC.

RETRACT : questo comando serve per ritirare le shared variables FR ed FC.

RESET : questo comando serve per cancellare i dati di telemetria che sono nello spazio di lavoro.

5. comandi per la lettura dei dati

READ : questo comando serve per leggere i dati dal file di telemetria su disco.

6. comandi per l'elaborazione

PLOT : tramite questo comando l'utente chiede che vengano esaminate delle grandezze di telemetria e ne venga fatto un grafico.

FLAG : tramite questo comando l'utente chiede che vengano cancellate dalla matrice FRAME tutte quelle major frames che contengono valori non 'buoni' per quelle grandezze che l'utente ha specificato nell'istruzione.

7. comandi per salvare lo spazio di lavoro

SAVE : questo comando scrive su disco tutto quello che e' contenuto nello spazio di lavoro attivo ad un certo istante.

8. comandi per terminare la seduta

END : con questo comando l'utente termina la seduta e puo' chiedere di restare in ambiente APL o in ambiente CMS. Lo spazio di lavoro viene salvato.

OUT : come il comando END. Lo spazio di lavoro pero' non viene salvato.

3. Uso del TPS

3.1 Istruzioni

L'utente comunica col sistema inviando delle opportune istruzioni costituite da un comando che puo' essere seguito o meno da operandi. La forma generale di una istruzione e' la seguente:

COMANDO <arg1> . . . <argn>

L'esecuzione di una istruzione avviene attraverso le seguenti fasi:

a. riconoscimento del comando

Il sistema controlla se il comando ricevuto e' contenuto nella lista dei comandi ammessi; se il confronto da' risultato positivo passa alla seconda fase, altrimenti dopo una segnalazione di errore si pone in attesa di una nuova istruzione.

b. controllo degli operandi

dopo aver riconosciuto il comando il sistema controlla che gli operandi siano validi per quel comando: in caso di errore l'istruzione viene rigettata, viene inviato un messaggio di errore e il sistema si pone in attesa di una nuova istruzione, altrimenti passa alla fase successiva.

c. elaborazione

dopo aver ricevuto il comando, gli operandi ed averne controllato la validita' il sistema e' pronto per eseguire le operazioni richieste. L'elaborazione termina con la stampa dei risultati e di una stringa (...) che indica che il sistema e' pronto a ricevere l'istruzione successiva.

3.2 Descrizione dei comandi

I nomi dei comandi possibili sono i seguenti:

CMS
COMDS
CP
END
FLAG
HELP
OUT
PLOT
READ
RESET
RETRACT
SAVE
SHARE

START
STATUS

Diamo di seguito una descrizione dettagliata dei vari comandi secondo il seguente schema:

- a. descrizione della sintassi del comando
- b. descrizione della semantica
- c. eventuali messaggi di errore

CMS <SAVE / NOSAVE>

Questo comando abilita il CMS dall'interno del sistema. Più precisamente, dopo aver dato questo comando, l'utente può inviare qualsiasi comando del CMS SUBSET. Per riprendere l'esecuzione deve inviare il comando RETURN. Per il significato delle possibili opzioni SAVE e NOSAVE vedere la descrizione del comando CP.

messaggi di errore: INCORRECT COMMAND

COMDS

questo comando permette all'utente di avere a terminale la lista dei comandi possibili con la descrizione del loro formato.

messaggi di errore: INCORRECT COMMAND

CP <SAVE / NOSAVE>

questo comando abilita il CP dall'interno del sistema. Gli argomenti possibili sono SAVE per salvare lo spazio di lavoro prima di entrare in ambiente CP oppure NOSAVE. Se non ci sono argomenti viene assunto per default il valore SAVE.

messaggi di errore: INCORRECT COMMAND

END <APL / CMS>

questo comando permette di uscire dal sistema per passare in ambiente APL o CMS a secondo dell'opzione scelta. Non è necessario che l'utente dia il comando SAVE perché il sistema TLM salva automaticamente lo spazio di lavoro. Se non vengono passati argomenti il sistema assume per default l'opzione APL.

messaggi di errore: INCORRECT COMMAND

FLAG OF <INF1 A/B> <INF2 A/B>

EIN A/B

SPF A/B

SVB A/B <{t1r1 t1r2}>

<(65 115)>

WID A/B <{t1r1 t1r2}>

<(11 19 >

questo comando controlla che la misura contenuta nel campo specificato nell'argomento abbia un valore accettabile. In caso contrario la major frame in questione non puo' essere usata per la determinazione dell'assetto e viene percio' eliminata dalla matrice FRAME. Questo comando richiede sempre un argomento. Vediamo in dettaglio i singoli casi:

QF puo' esser seguito da una o piu' opzioni che indicano in quali mjr frames va controllato il quality flag. Se QF non e' seguito da nessuna opzione il comando esegue il controllo su tutte le mjr frames.

EIN deve esser specificato il package. Il comando controlla che l'EARTH-IN sia diverso da zero.

WID deve essere specificato il package. Il comando controlla che la corda sia entro limiti opportuni. E' possibile specificare due valori entro cui si vuole avvenga il controllo. Se non e' dato alcun valore di tolleranza viene assunto per default l'intervallo <11 19> gradi.

SPF deve essere specificato il package. Il comando controlla che la SPIN RATE sia diversa da zero.

SVB deve essere specificato il package. Il comando verifica che il SUN ANGLE sia contenuto entro valori opportuni. E' possibile specificare i valori limite tramite un vettore a due componenti. Se non e' specificata alcuna tolleranza il comando assume per default l'intervallo <65 115> gradi.

HELP <READ / PLOT / FLAG / SHARE>

tramite questo comando l'utente puo' richiedere che venga stampato uno dei comandi sopra spacificati con la relativa sintassi. Se il comando HELP non e' seguito da argomenti vengono scritti tutti i comandi TPS e il loro formato. messaggi di errore: INCORRECT COMMAND

OUT <APL /CMS>

con questo comando l'utente termina la sessione ed esce dal sistema senza che lo spazio di lavoro venga salvato. Scegliendo l'opzione APL l'utente entra in ambiente APL mentre scegliendo l'opzione CMS passa in ambiente CMS. Se l'utente non specifica nessuna opzione il sistema assume per default il valore APL. messaggi di errore: INCORRECT COMMAND

PLOT QF <INF1 A/B> <INF2 A/B> <SPF A/B> <SVB A/B>
<ILFIN A/B> <ILFOUT A/B> <SUNS *> <EIN *>
TIME <FROM FR N1> <TO FR N2>
SPF A/B
INF1 A/B
INF2 A/B
SVB A/B
WIDTHS A/B

questo comando, che deve essere sempre seguito da un

argomento, fornisce la rappresentazione grafica dei dati richiesti. Vediamo in dettaglio i casi possibili:

TIME

con questa opzione compare sullo schermo il grafico dell'intervallo di tempo fra due frames consecutive della matrice FRAME. In ascissa c'è una scala numerata da N1 a N2. Se N1 e N2 non sono stati specificati nel comando allora N1 ed N2 sono rispettivamente la prima e l'ultima riga della matrice FRAME. Sempre in ascissa compare anche una scala dei tempi in HH.MM.SS. In ordinata la scala dei tempi è in DD. HH. MM. SS. Ciascun grafico è accompagnato dalle seguenti statistiche:

- a. numero di major frames plottate
- b. massimo intervallo di tempo fra due major frames consecutive
- c. minimo intervallo di tempo fra due major frames consecutive (millisec)
- d. intervallo di tempo medio tra due major frames consecutive (millisec)

QF

con questa opzione compare sullo schermo un grafico che ha in ascissa il numero progressivo delle major frames e in ordinata i valori logici 0 1 come risultato del controllo effettuato sul quality flag. All'interno di una major frame vengono verificati i quality flags delle mjinor frames indicate dall'utente nell'istruzione tramite quei parametri che sono specificati entro parentesi. L'utente può scrivere uno o più parametri di seguito, separati da almeno un blank. Se non viene specificato alcun parametro il comando controlla tutte le mjinor frames.

INF1/INF2

con questa opzione il risultato sullo schermo è un grafico che ha in ascissa il numero progressivo delle major frames e in ordinata le corrispondenti misure in gradi dei seguenti angoli:

SUN TO EARTH-IN
SUN TO EARTH-OUT

I valori selezionati possono essere relativi al package A o al package B e al sensore 1 o al sensore 2, come specificato nell'istruzione. Il comando fornisce anche le seguenti informazioni: tipo del package e campo di variazione delle funzioni plottate.

SPF

con questa opzione il risultato è un grafico che plotta i valori della SPIN RATE in gradi/sec. Il comando fornisce anche le seguenti informazioni: tipo del package, campo di variazione della SPIN RATE, valore medio.

SVB

con questa opzione vengono plottati i valori del SUN ANGLE in gradi. Il comando fornisce anche le

seguenti informazioni: tipo di package e campo di variazione del SUN ANGLE.

ILF

con questa opzione vengono plottati i valori del SUN TO EARTH-IN e del SUN TO EARTH-OUT che vengono trasmessi dai sensori a campo largo. Vengono date anche le seguenti informazioni:

- a. package a cui si riferiscono i valori plottati
- b. campo di variazione del SUN TO EARTH-IN
- c. campo di variazione del SUN TO EARTH-OUT

WIDTHS

con questa opzione viene disegnata la corda relativa al package specificato. In ordinata la scala e' in gradi, in ascissa c'e' una numerazione progressiva che si riferisce alle frames considerate e la scala dei tempi.

messaggi di errore: INCORRECT COMMAND
INVALID START RECORD NUMBER
INVALID END RECORD NUMBER
END RECORD NUMBER SET TO BOTTOM OF FRAME
NO FRAMES HAVE BEEN ALLOCATED

READ <FROM REC N1> <TO REC N2> <EVERY I REC> <(ADD)>
<FROM TIME N1> <TO TIME N2> <EVERY I SEC> <(REP)>

questo comando provoca la lettura dei records che si trovano su disco nel file di nome TLMTRY e di tipo FILE, la conversione dei dati dalla rappresentazione binaria a quella decimale e la concatenazione di ciascun record letto alla matrice FRAME. Se non ci sono opzioni i records vengono letti e aggiunti a quelli gia' esistenti in FRAME fino a che la matrice non ha raggiunto 250 righe. Con l'opzione REP la matrice FRAME viene azzerata prima che nuovi records vengano letti. Questo comando permette di leggere dei records dal file di telemetria in modo non sequenziale tramite le opzioni:

EVERY I REC
EVERY I SEC

dove I e' un intero. I valori N1 ed N2 che compaiono nell'istruzione possono essere degli interi o dei tempi: nel primo caso indicano rispettivamente il primo e l'ultimo record da leggere, nel secondo caso indicano il tempo del primo e dell'ultimo record da leggere. Se si specifica il tempo questo deve avere il seguente formato: hhmm.ss

messaggi di errore: INCORRECT COMMAND
INVALID START RECORD NUMBER
INVALID START TIME
INVALID RECORD SKIPPING
INVALID TIME SKIPPING
INVALID END RECORD NUMBER
INVALID END TIME
END RECORD NUMBER SET TO BOTTOM OF
TLMTRY FILE
MAX NUMBER OF FRAMES REACHED

RESET

questo comando, che non ammette opzioni, cancella le frames presenti in memoria e posiziona il puntatore di lettura sul primo record del file di telemetria.
messaggi di errore: INCORRECT COMMAND

RETRACT

questo comando, che non ammette operandi, ritrae dallo sharing le variabili spartite.
messaggi di errore: INCORRECT COMMAND

SAVE

questo comando non ammette operandi. Il risultato e' quello di salvare lo spazio di lavoro. Non si esce dal sistema.
messaggi di errore: INCORRECT COMMAND

SHARE <fname ftype fmode>

questo comando serve per inizializzare l'auxiliary processor 110 per la lettura del file di telemetria. Gli eventuali operandi rappresentano nome, tipo e modo del file di CMS che si vuole leggere. Se l'utente non fornisce argomenti viene letto il file 'TELEMETRY FILE B'.
messaggi di errore: INCORRECT COMMAND
SHARE ERROR
OPEN ERROR
EMPTY FILE

START

questo comando, che non richiede alcun argomento, serve ad inizializzare il sistema. Dopo che e' stato dato questo comando compare sullo schermo una stringa (...) che indica all'utente che il sistema e' pronto a ricevere il successivo comando.
messaggi di errore: SHARE ERROR

STATUS

questo comando, che non richiede opzioni, produce la stampa delle seguenti informazioni riguardanti la situazione della matrice FRAME:

NUMBER OF FRAMES IN CORE: N
FROM TIME HHMMSS - DD MONTH
TO TIME HHMMSS - DD MONTH

Se non ci sono frames in memoria viene stampato il seguente messaggio: NO FRAMES IN CORE
messaggi di errore: INCORRECT COMMAND

2.3 Uso del sistema

I passi necessari per accedere al sistema sono stati dettagliatamente descritti nel paragrafo 2.1.

La funzione principale dello spazio di lavoro TPS e' la START che deve essere chiamata all'inizio della seduta. Notare, pero', che se l'utente ha usato la procedura TPS per accedere al sistema, non e' piu' necessario inviare il comando START poiche' e' implicito nella procedura stessa.

La funzione START scrive sullo schermo il messaggio (...) e aspetta di ricevere un comando. Se la sintassi del comando e' errata manda un messaggio di errore e si pone in attesa di una nuova istruzione, altrimenti chiama la funzione opportuna che esegue il comando. Alla fine della esecuzione stampa ancora la stringa (...), si sblocca la tastiera, aspetta di ricevere in input la successiva istruzione e cosi' via fino a quando non vengono dati i comandi END o OUT di fine sessione.

L'utente puo' mandare un comando alla volta. Le richieste HELP o COMDS possono essere fatte in qualsiasi momento durante la seduta poiche' la loro esecuzione e' indipendente da qualsiasi altro tipo di elaborazione. La rappresentazione di grafici e' invece legata alla presenza nello spazio di lavoro di una matrice, FRAME, che contiene i dati di telemetria. Tramite il comando STATUS l'utente puo' verificare, ogni volta che lo ritiene opportuno, la situazione dei dati nello spazio di lavoro attivo. Normalmente la matrice FRAME contiene i dati letti nella piu' recente seduta durante la quale e' stato dato il comando SAVE o che e' terminata con il comando END. In entrambi i casi, infatti, lo spazio di lavoro e' stato salvato. Per l'aggiornamento dei dati e' disponibile il comando READ che esegue la lettura dei dati di telemetria da disco e li aggiunge alla matrice FRAME. E' importante notare che la richiesta di READ puo' essere fatta dall'utente solo dopo che sono state messe in sharing con i processors opportuni, tramite il comando SHARE, le variabili utilizzate dal sistema per la lettura dei dati di telemetria, altrimenti compare un messaggio di errore. Il comando SHARE va dato solo una volta durante una sessione e precisamente prima della prima richiesta di READ.

I dati possono essere elaborati tramite il comando PLOT o il comando FLAG. Va notato che PLOT cancella lo schermo prima di disegnare un nuovo grafico mentre tutti gli altri comandi non producono la cancellazione dello schermo. E' l'utente che deve percio' provvedere a cancellare lo schermo quando lo ritiene opportuno per evitare la sovrapposizione dei risultati.

Anche i comandi CP e CMS meritano particolare attenzione. Quando l'utente e' in uno di questi ambienti deve evitare di inviare comandi che possano provocare una brusca interruzione dell'API. Per esempio i comandi DEF STOR e I CMS provocano un risultato disastroso. Alcuni comandi di CMS e CP utilizzabili sono invece: ACCESS, FRASE, LISTFILE, SPOOL, QUERY, DETACH, LINK etc. In particolare con il comando LINK l'utente puo' richiedere l'accesso ad un minidisco diverso da quello che il sistema collega per default all'inizio della seduta.

Per terminare la seduta l'utente può usare sia il comando END che il comando OUT a seconda se vuole salvare oppure no lo spazio di lavoro attivo.

Tutti i comandi sopra citati sono descritti dettagliatamente nel paragrafo 3.2 dove è illustrata anche la sintassi di ciascun comando. Notare che i parametri ~~tra~~ parentesi sono opzionali; la parentesi rotonda aperta '(' è parte del parametro che la segue quindi va scritta; i parametri separati da '/' sono esclusivi.

2.4 Messaggi di errore

Sono previsti dei controlli per evitare un uso non corretto dei comandi. Diamo di seguito una descrizione dettagliata dei possibili messaggi di errore e del loro significato:

INCORRECT COMMAND: questo messaggio indica che è stato rilevato un errore di tipo sintattico nell'istruzione. Può essere sbagliato il nome del comando o quello di un operando. Dopo aver segnalato l'errore il sistema si dispone a ricevere una nuova istruzione.

REENTER: questo messaggio indica all'utente che il sistema è in attesa di ricevere un nuovo valore per un parametro fornito precedentemente in modo non esatto.

INVALID START TIME

INVALID END TIME

INVALID TIME SKIPPING

INVALID START RECORD NUMBER

INVALID END RECORD NUMBER

INVALID RECORD SKIPPING: questi messaggi indicano all'utente che un parametro è stato specificato in modo inesatto. Può essere stato dato un parametro non numerico o con un valore al di fuori del range previsto. Ogni volta che compare uno di questi messaggi, il sistema invia successivamente la richiesta REENTER che è stata descritta sopra.

END RECORD NUMBER SET TO BOTTOM OF TELEMETRY FILE: questo messaggio segnala all'utente che l'ultimo record letto da disco è anche l'ultimo record del file di telemetria. Non c'è interruzione dell'esecuzione.

MAX NUMBER OF FRAMES REACHED: questo messaggio segnala all'utente che è stato occupato tutto lo spazio di memoria disponibile per i dati da elaborare. In questo caso l'esecuzione continua ma non possono essere lette altre frames se prima non vengono cancellate quelle già allocate. Prima di dare il comando di lettura l'utente deve inviare il comando RESET oppure specificare l'opzione (REP nel comando READ.

NO FRAMES HAVE BEEN ALLOCATED: questo messaggio segnala all'utente che in memoria non ci sono dati. Il sistema interrompe l'elaborazione e si dispone a ricevere nuove istruzioni. L'utente, prima di procedere a qualsiasi

elaborazione, deve inviare il comando di lettura READ.

END RECORD NUMBER SET TO BOTTOM OF FRAME: questo messaggio compare se l'utente chiede che vengano elaborate un numero di frames maggiore di quelle presenti in memoria. In questo caso l'esecuzione continua e il messaggio segnala solo che l'ultima frame elaborata e' l'ultima frame presente in memoria.

SHARE ERROR: questo messaggio informa l' utente che non sono state messe in sharing le variabili necessarie per l' operazione di lettura. Dopo questa segnalazione il sistema si dispone a ricevere un nuovo comando. L' utente puo' inviare il comando SHARE e successivamente il comando READ.

OPEN ERROR: questo messaggio compare quando il valore di inizializzazione di una variabile che si vuole spartire non e' accettato dal processor. In questo caso il sistema interrompe l' elaborazione e passa il controllo all' APL.

EMPTY FILE: questo messaggio puo' comparire dopo che l' utente ha dato il mando SHARE e sta ad indicare che il file di telemetria su disco e' vuoto. In questo caso il sistema interrompe l'elaborazione e passa il controllo all' APL.

4. Bibliografia

- 1) APL language, File No. S370-22, Order No. GC26-3847-1
- 2) VS APL for CMS: Terminal User's Guide
File No. SH20-9067-1, Program Number 5748-AP1
- 3) IBM Virtual Machine Facility/370: CMS User's Guide,
File No. S370-39, Order No. GC20-1819-0
- 4) SIRIO S/C Operational Manual, CIA, Roma, Marzo 1977
- 5) SIRIO Attitude Determination Program (ADP),
Manuale d' uso,
Faconti, Lucchesi, Pasquinelli,
CNUCE, Pisa, Luglio 1977
- 6) SIRIO Telemetry Processor, User' s Guide,
Contract NAS 5-11999, CSC,
Greenbelt, Maryland, Dicembre 1976
- 7) The flight Dynamics System For The Control Of The Sirio
S/C During Its Operational Life,
Faconti, Trumpy,
CNUCE, Pisa, Febbraio 1977

APPENDICE A

Listings delle funzioni

```

    ▽ Z←ADD X
[1]  Z←0
    ▽
    ▽ ASSI CART;XX;YY;PA
[1]  XX←XSC
[2]  YY←YSC
[3]  OUTV TKX 3 2 ρCART
[4]  LOOPX:→(0=ρXX)/LOOPY
[5]  OUTV TKX 2 2 ρXX[1],250,XX[1],240
[6]  XX←1+XX
[7]  →LOOPX
[8]  LOOPY:→(0=ρYY)/SC
[9]  OUTV TKX 2 2 ρ490,YY[1],500,YY[1]
[10] YY←1+YY
[11] →LOOPY
[12] SC:(((1+ρYSTR),2)ρGS),(((1+ρYSTR),4)ρTKX((1+ρYSTR)ρ490-
    (-1+ρYSTR)×8),[1.1]YSC),(((1+ρYSTR),2)ρUS),YSTR
[13] XSC←XSC-30
[14] (((ρXSTR),2)ρGS),(((ρXSTR),4)ρTKX(((ρXSTR),1)ρXSC),((ρ
    XSTR),1)ρ220),(((ρXSTR),2)ρUS),((ρXSTR),4)ρ 4 0 ρXSTR
[15] (((ρXSTR),2)ρGS),(((ρXSTR),4)ρTKX(((ρXSTR),1)ρXSC-15),
    ((ρXSTR),1)ρ200),(((ρXSTR),2)ρUS),TIM
    ▽
    ▽ Z←BEAM;Z1
[1]  Z←(1+ ' ')<=>]vΛ≠÷,+. /0123456789(Γ;x:\^_α|n|ε_∇Δι°' '□]
    τo*?ρ[~↓uω▷†c←→≥-ABCDEFGHIJKLMNOPQRSTUVWXYZ' iZ1←5†
    CBEAM
[2]  Z←(Z[3]+32×Z[2]),Z[5]+32×Z[4]
    ▽
    ▽ Z←CN CALINF C;I;SPR;EW
[1]  →(Z←'*=1+C)/0
[2]  →(Z←2<I← 'AB' i1+C)/0
[3]  →( 1 2 =I)/A,B
[4]  Z←1
[5]  →0
[6]  A:I← 0 9
[7]  SPR←,FRAME[; 34 80 ]÷65536
[8]  →EX
[9]  B:I← 1 8
[10] SPR←,FRAME[; 35 79 ]÷65536
[11] EX:SPR[(0≠SPR)/iρSPR]←360÷SPR[(0≠SPR)/iρSPR]
[12] →(CN=1)/W
[13] EW←,FRAME[;4+I×5]<FRAME[;5+I×5]
[14] EIN←SPR×(,FRAME[; 55 55 ]-FRAME[; 54 54 ]):65536
[15] EIN[X]←360+EIN[X←(EIN<0)/iρEIN]
[16] EIN←EIN×ε 'EW EW' [ 0 3 [CN-3]+i3]
[17] EOUT←EIN+(,FRAME[;CN+I×5]):65536)×SPR

```


[18] →0
[19] W: EIN←SPR×, FRAME[; 4+I×5] ÷ 65536
[20] EOUT←SPR×, FRAME[; 5+I×5] ÷ 65536

▽

▽ Z←CALQF A; PARM

[1] PARM←A
[2] PARM←PARM[ΔPARM]
[3] PARM←(PARM≠1φPARM) / PARM
[4] Z←(0≠PARM) / PARM

▽

▽ Z←CALSUN C; I; SPR

[1] →(Z←' * ' = 1 + C) / 0
[2] →(Z + 1 ≠ pC) / 0
[3] →(Z + 2 < I ← ' AB ' 1 + C) / 0
[4] →(1 2 = I) / A, B
[5] A: FUN←FRAME[; 15 59] ÷ 65536
[6] SPR←, FRAME[; 34 80] ÷ 65536
[7] →EX
[8] B: FUN←FRAME[; 14 60] ÷ 65536
[9] SPR←, FRAME[; 35 79] ÷ 65536
[10] EX: SPR[(0≠SPR) / 1pSPR] ← 360 ÷ SPR[(0≠SPR) / 1pSPR]
[11] FUN←(, FUN) × SPR
[12] FUN←(3010FUN × 0 ÷ 180) × 180 ÷ 01

▽

▽ Z←CBEAM

[1] ' ωFω7 '
[2] Z←[]
[3] [AV [203]

▽

▽ Z←CKENDRD T

[1] →(250 ≤ 1 + pFRAME) / EOF
[2] →(EN) / ET
[3] →(EP < FC [2]) / Z ← 0
[4] Z ← 1
[5] → 0
[6] ET: →(∧ / EP < , 0 40 CONV T) / Z ← 0
[7] Z ← 1
[8] → 0
[9] EOF: ' MAX NUMBER OF FRAMES REACHED '
[10] Z ← 0

▽

```

▽ Z←CKIT T;T1;X
[1] →(Λ/ 4 3 ≠ρT1←(¯1+X←(,T)ι'.' )†T)/ERR
[2] T←PACK T
[3] →(6<ρT←X+T)/ERR
[4] Z←(⊂2†T1),⊂2†T1
[5] →(12<1†Z)/ERR
[6] →((X← 31 28 31 30 31 30 31 31 30 31 30 31. )†1†Z]†1†Z)/
ERR
[7] →(0=ρT)/DATE
[8] →(24<¯1†Z←Z,⊂2†T)/ERR
[9] →(0=ρT←2†T)/DATE
[10] →(60<¯1†Z←Z,⊂2†T)/ERR
[11] →(0=ρT←2†T)/DATE
[12] →(60<¯1†Z←Z,⊂2†T)/ERR
[13] →(√/0≤Z)/DATE
[14] ERR:→Z←0
[15] DATE:Z←((+¯1+0,X)†1†Z]†Z[2]-1),4†2†Z
[16] Z← 365 24 60 60 1000 1Z

```

▽

```

▽ Z←CKQF P;PARM;I;X;PR;V;TYP
[1] V←((1†ρSENS),2)ρ 1 10 2 9 1 10 2 9 7 16 7 16 3 12 3
12 6 13 5 14 6 13 5 14 11 0 11 0
[2] PARM←ι0
[3] LOOP:→(7<ρPR←(¯1+X←Pι' ')†P)/ERR
[4] →(0=ρP←X+P)/ERR
[5] →(' *AB' Λ.≠1†TYP←(¯1+X←Pι' ')†P)/ERR
[6] I←(SENSΛ.=8†PR,' ',TYP)ι1
[7] →(I>1†ρSENS)/ERR
[8] PARM←PARM,V[I;]
[9] →(0≠ρP←X+P)/LOOP
[10] Z←PARM
[11] →0
[12] ERR:Z←1

```

▽

```

▽ Z←CKREAD P;OP;X;PR1;PR2;I;PR3;TCOM;TPARM;C
[1] Z←0
[2] TCOM← 4 9 ρ' '
[3] TPARM← 4 1 ρ' '
[4] OP←PACK 1†(X←¯1+(,P)ι'(' )†P
[5] →(0=ρP←PACK X+P)/OPT
[6] LOOP:→(5<ρPR1←(¯1+X←Pι' ')†P)/ERR
[7] →( 1 2 3 =I←(RDSUBΛ.=5†PR1)ι1)/P1,P1,P3
[8] →ERR
[9] P1:→(5<ρPR2←(¯1+X←Pι' ')†P←X+P)/ERR
[10] →( 4 5 9 =(RDSUBΛ.=5†PR2)ι1)/P11,P12,P11
[11] →ERR
[12] P11:TCOM[I;]←¯9† ' ' ,PR1, 'RECA'
[13] →P2

```

```

[14] P12:TCOM[I;]←-9↑ ' ' ,PR1, 'TIME'
[15] P2:PR3←(-1+X←P↑ ' ')↑P←X↓P
[16] P4:TPARM←((1↑ρTPARM),C←(-1↑ρTPARM)↑ρPR3)↑TPARM
[17] TPARM[I;]←C↑PR3
[18] →(0≠ρP←X↓P)/LOOP
[19] →OPT
[20] P3:PR3←(-1+X←P↑ ' ')↑P←X↓P
[21] →(5<ρPR2←(-1+X←P↑ ' ')↑P←X↓P)/ERR
[22] →( 4 8 9 =(RDSUBΛ.=5↑PR2)↑1)/P31,P32,P31
[23] →ERR
[24] P31:TCOM[I;]←-9↑PR1, 'RECA'
[25] →P4
[26] P32:TCOM[I;]←-9↑PR1, 'TIME'
[27] →P4
[28] OPT:→(0=ρOP)/TEST
[29] →(5<ρPR1←(-1+X←OP↑ ' ')↑OP)/ERR
[30] →(Λ/ 6 7 ≠(RDSUBΛ.=5↑PR1)↑1)/ERR
[31] →(0≠ρOP←X↓OP)/ERR
[32] TCOM[4;]←9↑PR1
[33] TEST:→( 0 3 =ρX←(C←( 3 9 ↑TCOM)Λ.= ' ')/ 1 2 3 )/T0,T3
[34] TCOM[X;]←( 3 5 ρ ' FROM TOEVERY' )↑X;],TCOM(ρX)ρ1↑(
~C)/ 1 2 3 ; 6 7 8 9 ]
[35] TPARM[X;]←((ρX),-1↑ρTPARM)↑((ρX),1)ρ'*'
[36] →T0
[37] T3:TCOM[X;]← 3 9 ρ 'FROMRECA TORECA'
[38] TPARM[X;]←(3,-1↑ρTPARM)↑ 2 1 ρ'*'
[39] T0:→(√/√/XΛ.≠QX← 3 -4 ↑TCOM)/ERR
[40] Z←TCOM,TPARM
[41] →0
[42] ERR:Z←1

```

▽

▽ Z←CMS A

```

[1] AQUESTA FUNZIONE ABILITA IL CMS DALL'INTERNO DEL
SISTEMA
[2] APOSSIBILI ARGOMENTI SONO '***' E 'SAVE' (DEFAULT) PER
[3] ASALVARE LO WS PRIMA DI ABILITARE IL CMS, OPPURE '
NOSAVE'
[4] Z←1
[5] →(0≠CPCMS A)/0
[6] SV100← 'CMS'
[7] SV100← 'SUBSET'
[8] Z←0

```

▽

▽ COMDES

```

[1] '-COMDS'
[2] '-CMS [SAVE] [NOSAVE]'
[3] '-CP [SAVE] [NOSAVE]'
[4] '-END [APL] [CMS]'

```

```

[5]  '-OUT      [APL] [CMS]'
[6]  '-PLOT     QF      [INF1 A/B] [INF2 A/B] [SPF A/B] [
SVB A/B]'
[7]  '          [ILFIN A/B] [ILFOUT A/B] [SUNS *] [
EIN *]'
[8]  '          [SUNS *] [EIN *]'
[9]  '          TIME    [FROM FR N1] [TO FR N2] '
[10] '          SPF      A/B'
[11] '          INF1     A/B'
[12] '          INF2     A/B'
[13] '          SVB      A/B'
[14] '          ILF      A/B'
[15] '          WIDTHS   A/B'
[16] '-READ     [FROM REC N1][TO REC N2][EVERY I REC][
ADD]'
[17] '          [FROM TIME T1][TO TIME T2][EVERY I SEC][
REP]'
[18] '-RESET
[19] '-SAVE
[20] '-STATUS'
[21] '-FLAG     QF      [INF1 A/B] [INF2 A/B] [SPF A/B] [
SVB A/B]'
[22] '          [ILFIN A/B] [ILFOUT A/B] [SUNS *] [
EIN *]'
[23] '          EW      A/B [( 11  19 )]'
[24] '          [(TLR1 TLR2)]'
[25] '          SVB     A/B [( 65  115 )]'
[26] '          [(TLR1 TLR2)]'
[27] '          EIN     A/B'
[28] '          SPF     A/B'
[29] '          ILF     A/B'
[30] '-HELP     [READ / PLOT / FLAG / SHARE]'
[31] '-SHARE    [FNAME FTYPE FMODE]'

```

▽

▽ Z←COMDS A

```

[1]  QUESTA FUNZIONE STAMPA AL TERMINALE IL CONTENUTO
      DELLA VARIABILE
[2]  COMDES CONTENENTE LA DESCRIZIONE DEI COMANDI
[3]  Z←1
[4]  →(Λ/ '***' ≠3↑A)/0
[5]  COMDES
[6]  Z←0

```

▽

▽ Z←I CONV T

```

[1]  Z← 16 1 p21QT[( -144+I[1]+144×i16)°.+iI[2]]

```

▽

▽ Z←CONVERT T

[1] Z←(0 40 CONV T),(40 8 CONV T),(48 9 CONV T),(56
16 CONV T), 72 16 CONV T

▽

▽ Z←CP A

[1] QUESTA FUNZIONE ABILITA IL CP DALL'INTERNO DEL SISTEMA

[2] POSSIBILI ARGOMENTI SONO '***' E 'SAVE' (DEFAULT) PER

[3] SALVARE LO MS PRIMA DI ABILITARE IL CP, OPPURE '
NOSAVE'

[4] Z←1

[5] →(0≠CPCMS A)/0.

[6] SV100← 'CP'

[7] Z←0

▽

▽ Z←CPCMS A

[1] QUESTA FUNZIONE FA IL CHECK DEGLI ARGOMENTI PASSATI

[2] ALLE FUNZIONI CP E CMS. SE SONO '***' O 'SAVE'

[3] SALVA LO SW.

[4] →(3 4 6 =p,A)/C34,C34,C6

[5] ERR:Z←1

[6] →0

[7] C34:→(V/(2 4 p '*** SAVE')∧.=4+A)/SA

[8] →ERR

[9] C6:→(∧/ 'NOSAVE' ≠A)/ERR

[10] →Z←0

[11] SA:Z←SAVE '***'

[12] Z←0

▽

▽ Z←END A

[1] QUESTA FUNZIONE SALVA LO MS ED ESCE DAL SISTEMA

[2] POSSIBILI ARGOMENTI SONO: '***' E 'APL' (DEFAULT) PER
RESTARE IN APL

[3] OPPURE 'CMS' PER PASSARE IN CMS

[4] Z←SAVE '***'

[5] Z←OUT A

▽

▽ Z←EVERYRECA X
 [1] EN←0
 [2] A:→('*'≠1↑X)/C
 [3] SKIP←1
 [4] →Z←0
 [5] C:→(∼^/Xε ' 0123456789')/ERR
 [6] SKIP←2X
 [7] →Z←0
 [8] ERR: 'INVALID RECORD SKIPPING'
 [9] →(0≠ρ,X←REENTER)/A
 [10] Z←1

▽

▽ Z←EVERYTIME X
 [1] EN←1
 [2] A:→('*'≠1↑X)/C
 [3] SKIP←16
 [4] →Z←0
 [5] C:→(∼^/(PACK X)ε '0123456789')/ERR
 [6] SKIP←2X
 [7] →Z←0
 [8] ERR: 'INVALID TIME SKIPPING'
 [9] →(0=ρ,X←REENTER)/A
 [10] Z←1

▽

▽ Z←FLAG AA;TAB;PARM;OP;X;TLR;I;SEL;EIN;EOUT;FUN;TIB,
 SPR;DIM;ZZ
 [1] →(Z←^/ '***' =3↑AA)/0
 [2] DIM←1↑ρFRAME
 [3] TAB← 6 4 ρ 'QF EIN EW SVB ILF SPF '
 [4] TIB← 4 6 ρ 'INF1 AINF2 AINF1 BINF2 B '
 [5] OP←(¯1+X←(AA=' ')i1)↑AA
 [6] AA←X+AA
 [7] TLR←(X←(AA=' ')i1)↑AA
 [8] PARM←PACK(¯1+X)↑AA
 [9] SU:→(Z←∼^/TLRε ' 0123456789.')/ER
 [10] →(0=ρTLR)/POI
 [11] TLR←2↑2TLR
 [12] →(>/TLR)/ER
 [13] POI:→(Z←6<X←(TAB^.=4↑OP)i1)/0
 [14] →((i6)=X)/QQ,IN,WID,SUN,IF,SPIN
 [15] Z←1
 [16] →0
 [17] QQ:ZZ←i16
 [18] →(Z←0≠ρTLR)/0
 [19] →(0=ρPARM)/TUT
 [20] →(0=ρρZ←CKQF PARM)/0
 [21] ZZ←CALQF Z
 [22] TUT:ZZ←¯3+5×ZZ

```

[23] FRAME←FRAME[(^/FRAME[;ZZ]=255)/i1↑pFRAME;]
[24] →DISP
[25] IN:FRAME←(0≠-/FRAME[; 55 54 ])/[1]FRAME
[26] →DISP
[27] SUN:→(0≠Z←CALSUM C←PARM)/0
[28] FUN←90-((1↑pFRAME),2)ρFUN
[29] →(0≠ρTLR)/E
[30] TLR← 65 115
[31] E:FRAME←(∼(∨/FUN≤TLR[1])∨(∨/FUN≥TLR[2]))/FRAME
[32] →DISP
[33] WID:→(Z←1≠ρPARM)/0
[34] →(Z←1 CALINF PARM)/0
[35] EIN←((1↑pFRAME),2)ρEIN
[36] EOUT←((1↑pFRAME),2)ρEOUT
[37] →(0≠ρTLR)/E2
[38] TLR← 11 19
[39] E2:EIN←∼∨/(EIN≤TLR[1])∨EIN≥TLR[2]
[40] EOUT←∼∨/(EOUT≤TLR[1])∨EOUT≥TLR[2]
[41] FRAME←(EIN∧EOUT)/FRAME
[42] →DISP
[43] IF:→(Z←0≠ρTLR)/0
[44] →(Z←1≠ρPARM)/0
[45] →( 1 2 =( 'AB' =PARM) i1)/IFA,IFB
[46] Z←1
[47] →0
[48] IFA:FRAME←(∼∧/0=FRAME[; 29 30 64 65 ])/FRAME
[49] →DISP
[50] IFB:FRAME←(∼∧/0=FRAME[; 24 25 69 70 ])/FRAME
[51] →DISP
[52] SPIN:→(Z←1≠ρPARM)/0
[53] X←,( 2 2 ρ 34 80 35 79 )[( 'AB' =1↑PARM)/i2;]
[54] FUN←FRAME[;X]÷65536
[55] FRAME←(∼∨/0=FUN)/FRAME
[56] DISP:PARM←OP,' ',PARM
[57] 'CHECKED FIELDS' ,((38-14+ρPARM)ρ' '),PARM
[58] (34+ 'NUMBER OF FRAMES FLAGGED ' ), 4 0 ρDIM-1↑ρ
FRAME
[59] →(0=ρTLR)/0
[60] (30+ 'TOLERANCE LIMITS' ), 4 0 ρTLP
[61] →0
[62] ER:TLR←REENTER
[63] →SU

```

∇

∇ Z←FROMRECA X

```

[1] EN←Z←0
[2] A:→(' '*≠1↑X)/C
[3] →('T'=-1↑X)/T1
[4] SP←1↑1+FC
[5] →0

```

```

[6] C:→('T'='1+X)/T2
[7] →(¬/X∈ ' 0123456789' )/ERR
[8] SP←FC←X
[9] →(≥/1+3+FC)/ERR
[10] →0
[11] ERR: 'INVALID START RECORD NUMBER'
[12] →(0≠ρ, X←REENTER)/A
[13] Z←1
[14] →0
[15] T1:SP←1
[16] →0
[17] T2:→(¬/(¬1+X)∈ ' 0123456789' )/ERT
[18] SP←X1+X
[19] →(SP≤1+ρFRAME)/0
[20] ERT: 'INVALID START RECORD NUMBER'
[21] →(0≠ρ, X←REENTERFR)/A
[22] Z←1

```

▽

▽ Z←FROMTIME X

```

[1] FN←Z←1
[2] A:→('*'≠1+X)/C
[3] SP←0
[4] FC←1
[5] →WNG
[6] C:→(0=SP←CKIT X)/ERR
[7] DOWN:X←LEGGI
[8] →(0≠1+FC)/0
[9] →(¬/SP≤, 0 40 CONV X)/UP
[10] →DOWN
[11] UP:→(1≥FC←FC[2]-2)/WNG
[12] X←LEGGI
[13] →(0≠1+FC)/0
[14] →(¬/SP≤, 0 40 CONV X)/UP
[15] →Z←0
[16] ERR: 'INVALID START TIME'
[17] FC←0
[18] →(0≠ρ, X←REENTER)/A
[19] →0
[20] WNG: 'READING STARTS FROM TOP OF TIMPY FILE'
[21] Z←0

```

▽

▽ Z←HELP H;CM;YY;TT;TT1;TT2

```

[1] Z←0
[2] TT← 4 5 ρ 'READ PLOT FLAG SHARE'
[3] TT1←1+ρTT
[4] TT2←¬1+ρTT
[5] →(¬/ '***' =3+H)/CONTR
[6] Z←COMDS 3+H

```



```

[7]   →0
[8]   CONTR:CM←(←1+(H=' ')i1)↑H
[9]   →(TT2<ρCM)/ERR
[10]  →(TT1<YY←(TT↑.=TT2↑CM)i1)/EPP
[11]  →( 1 2 3 4 =YY)/R,P,F,S
[12]  P: '-PLOT      OF      [INF1 A/B] [INF2 A/B] [SPF A/B] [
      SVE A/B]'
[13]  '           [ILFIN A/B] [ILFOUT A/B] [SUNS *] [
      EIN *]'
[14]  '           TIME [FROM FR N1] [TO FR N2] '
[15]  '           SPF   A/B'
[16]  '           INF1  A/E'
[17]  '           INF2  A/B'
[18]  '           SVB   A/B'
[19]  '           ILF   A/B'
[20]  '           WIDTHS A/P'
[21]  →0
[22]  R: '-READ      [FROM REC N1][TO REC N2][EVERY I REC][
      (ADD)']
[23]  '           [FROM TIME T1][TO TIME T2][EVERY I SEC][
      REP]'
[24]  →0
[25]  F: '-FLAG      OF      [INF1 A/B] [INF2 A/B] [SPF A/B] [
      SVB A/B] '
[26]  '           [ILFIN A/P] [ILFOUT A/B] [SUNS *] [
      EIN *]'
[27]  '           EW    A/B [( 11  19 ]'
[28]  '           [(TLP1 TLP2]'
[29]  '           SVB   A/B [( 65  115]'
[30]  '           [(TLP1 TLP2]'
[31]  '           EIN   A/P'
[32]  '           SPF   A/P'
[33]  '           ILF   A/B'
[34]  →0
[35]  S: '           SHARE [FNAME ETYPE FMODE]'
[36]  →0
[37]  ERR:Z←1

```

▽

▽ Z←ILF C;I;SPR;IN;OUT;YY;Y;X;XSTR;YSTR;XSC;YSC;RIG;TIM

```

[1]  RIG←,1
[2]  →(Z←'*'=1↑C)/0
[3]  →(Z←2<I← 'AB' i1↑C)/0
[4]  →( 1 2 =I)/A,B
[5]  A:I← 5 12
[6]  SPR←,FRAME[; 34 80 ]:=65536
[7]  →EX
[8]  B:I← 4 13
[9]  SPR←,FRAME[; 35 79 ]:=65536
[10] EX:SPF[(0≠SPR)/iρSPR]←360÷SPF[(0≠SPR)/iρSPR]

```

```

[11] YY←(FRAME[;(4+I×5),5+I×5]÷65536)
[12] IN←(,YY[; 1 2 ]×SPR
[13] OUT←(,YY[; 3 4 ]×SPR
[14] →(0≠Z←0 PLT IN,[0.1]OUT)/0
[15] YSTR←∇((ρ,YSTR),1)ρ,YSTR
[16] TIM← 3 0 ∇∇ 1 0 + 24 60 60 1000 τ(,FRAME[;1+I×5])[
XSTR]
[17] TIM[; 4 7 ]←((1↑ρTIM),2)ρ'. '
[18] ASSI 500 700 500 250 1000 250
[19] LOOP:→(0=1↑ρY)/ST
[20] OUTV TKX X,[1.1]Y[1;]
[21] Y← 1 0 +Y
[22] →LOOP
[23] ST:GS,(TKX 1 2 ρ 700 180 ),US, 'COUNTS'
[24] GS,(TKX 1 2 ρ 700 160 ),US, 'TIMES (HH.MM.SS)'
[25] ST:GS,(TKX 1 2 ρ 0 600 ),US, 'PACKAGE ' ,C
[26] GS,(TKX 1 2 ρ 0 550 ),US, 'ILFIN RANGE ' ,(∇ 10 2 ∇(
[/IN],[/IN], ' DEG'
[27] GS,(TKX 1 2 ρ 0 500 ),US, 'ILFOUT RANGE ' ,(∇ 10 2 ∇(
[/OUT],[/OUT], ' DEG'

```

∇

∇ Z←CN INF C;I;EIN;EOUT;X;Y;XSC;YSC;XSTR;YSTR;TIM;RIC

```

[1] RIC←,1
[2] →(0≠Z←CN CALINF C)/0
[3] LL:→(0≠Z←0 PLT EOUT,[0.1]EIN)/0
[4] YSTR←∇((ρ,YSTR),1)ρ,YSTR
[5] TIM← 3 0 ∇∇ 1 0 + 24 60 60 1000 τ,FRAME[; 1 1 ]
[6] TIM[; 4 7 ]←((1↑ρTIM),2)ρ'. '
[7] TIM←TIM[XSTR;]
[8] ASSI 500 700 500 250 1000 250
[9] LOOP:→(0=1↑ρY)/ST
[10] OUTV TKX X,[1.1]Y[1;]
[11] Y← 1 0 +Y
[12] →LOOP
[13] ST:GS,(TKX 1 2 ρ 0 500 ),US, 'PACKAGE ' ,C
[14] GS,(TKX 1 2 ρ 0 450 ),US, 'EIN RANGE ' ,(∇ 10 2 ∇([/
EIN],[/EIN], ' DEG'
[15] GS,(TKX 1 2 ρ 0 400 ),US, 'EOUT RANGE ' ,(∇ 10 2 ∇([/
EOUT],[/EOUT], ' DEG'

```

∇

▽ Z←INF1 C;CN

```
[1] CN←4
[2] →(0≠Z←CN INF C)/0
[3] GS,(TKX 1 2 ρ 350 650 ),US, 'INF1'
[4] GS,(TKX 1 2 ρ 350 630 ),US, '(DEG)'
[5] GS,(TKX 1 2 ρ 700 180 ),US, 'COUNTS'
[6] GS,(TKX 1 2 ρ 700 160 ),US, 'TIMES (HH.MM.SS.)'
```

▽

▽ Z←INF2 C;CN

```
[1] CN←5
[2] →(0≠Z←CN INF C)/0
[3] GS,(TKX 1 2 ρ 350 650 ),US, 'INF2'
[4] GS,(TKX 1 2 ρ 350 630 ),US, '(DEG)'
[5] GS,(TKX 1 2 ρ 700 180 ),US, 'COUNTS'
[6] GS,(TKX 1 2 ρ 700 160 ),US, 'TIMES (HH.MM.SS.)'
```

▽

▽ Z←LEGGI

```
[1] Z←192+FR
[2] →(0=1+FC)/0
[3] 'ERROR N. ' ,(▽1+FC), ' READING TLMTRY FILE'
```

▽

▽ Z←OUT X

```
[1] Z←1
[2] →(√/( 2 3 ρ '***APL' )∧.=3+X)/APL
[3] →(∧/ 'CMS' =3+X)/CMS
[4] →0
[5] APL:⊘'→'
[6] CMS:SV101← 'HT'
[7] SV101← ' )OFF HOLD'
[8] SΔOUT←OFF
[9] OFF:''
```

▽

▽ OUTV A;Z1;Z2;□PW

```
[1] □PW←255
[2] Z1←10
[3] L1:→(120≥ρA)/L2
[4] GS,Z1,Z2←120+A
[5] Z1← 4+Z2
[6] A←120+A
[7] →L1
[8] L2:GS,Z1,A,US
```

▽

▽ Z←PACK S;X
 [1] QUESTA FUNZIONE COMPATTA A 1 I BLANCK CONSECUTIVI IN
 UNA
 [2] STRINGA DI INPUT
 [3] Z←1+(~XΛ⁻¹ΦX←' '=S)/S←S, ' '
 ▽

▽ Z←PLOT OP;COM;X;CTC;RTC;ARG;TBC;I;A
 [1] 'ωFωE'
 [2] TBC←BEAM
 [3] →(2≠[NC 'FRAME'])/ATTN
 [4] →((0=1↑ρFRAME)∨80≠⁻¹↑ρFRAME)/ATTN
 [5] TBC← 9 6 ρ 'TIME QF INF1 INF2 ILF SVB SPF
 WIDTHS'
 [6] CTC←⁻¹↑ρTBC
 [7] RTC←1↑ρTBC
 [8] →(Z←Λ/ '***' =3↑OP)/0
 [9] COM←(⁻¹↑X←(OP=' ')i1)↑OP
 [10] →(Z←CTC<ρCOM)/0
 [11] →(Z←RTC<I←(TBCΛ.=CTC↑COM)i1)/0
 [12] L:→(0≠ρ,ARG←X↑OP)/EXEC
 [13] ARG←'*'
 [14] EXEC:Z←±COM, ' ', 'ARG'
 [15] →0
 [16] ATTN: 'NO FRAMES HAVE BEEN ALLOCATED'
 [17] Z←2
 ▽

▽ Z←FLAG PLT M;MM;ΔY
 [1] Z←0
 [2] →(Z←2≠ρρM)/0
 [3] MM← 1 0
 [4] →(FLAG=1)/POI
 [5] MM←(1+L0.5+Γ/Γ/M),(L0.5+L/L/M)-1
 [6] MM[(MM<0)/i2]←0
 [7] POI:Y←L0.5+250+(450÷ΔY←-/MM)×M-MM[2]
 [8] X←L0.5+500+(500÷(⁻¹↑ρY)-1)×0,i(⁻¹↑ρY)-1
 [9] →(5<ρX)/L1
 [10] XSC←X
 [11] XSTR←RIG[1]+0,⁻¹↑ρX
 [12] →L5
 [13] L1:XSC←500+100×0,i5
 [14] XSTR←(RIG[1]-1)+1,L0.5+((ρX)÷5)×i5
 [15] L5:→(5<ΔY)/L3
 [16] YSC←250+L0.5+(450÷ΔY)×0,iΔY
 [17] YSTR←MM[2]+0,iΔY
 [18] →CONST
 [19] L3:YSC←250+(450÷5)×0,i5
 [20] YSTR←L0.5+MM[2]+(ΔY÷5)×0,i5
 [21] CONST:→(1<1↑ρM)/0

[22] →(FLAG=0)/0
 [23] →((L/Y)≠[Y])/0
 [24] Y←(1, 1↑ρY)ρ450
 [25] YSC←,450
 [26] YSTR←,M[1;1]

▽

▽ Z←QF P;X;Y;XSC;YSC;XSTR;YSTR;GOOD;ZZ;RIC;TIM

[1] Z←0
 [2] RIF←,1
 [3] ZZ←,16
 [4] →('*=1↑P)/EX
 [5] →(0=ρρZ←CKQF P)/0
 [6] ZZ←CALQF Z
 [7] EX:ZZ←,3+5×ZZ
 [8] TIM← 3 0 ρ 1 0 ↓ 24 60 60 1000 τ,FRAME[;1]
 [9] TIM[; 4 7]←((1↑ρTIM),2)ρ'.
 [10] GOOD←Λ/255=FRAME[;ZZ]
 [11] →(0≠Z←1 PLT(1,ρGOOD)ρGOOD)/0
 [12] YSTR←ϕ((ρ,YSTR),1)ρ,YSTR
 [13] TIM←TIM[XSTR;]
 [14] ASSI 500 700 500 250 1000 250
 [15] OUTV TKX X,[1.1]Y[1;]
 [16] GS,(TKX 1 2 ρ 450 650),US, 'QF'
 [17] GS,(TKX 1 2 ρ 725 185),US, 'FRAMES'
 [18] GS,(TKX 1 2 ρ 725 170),US, 'TIMES (HH.MM.SS)'
 [19] GS,(TKX 1 2 ρ 0 550),US, 'NUMBER OF MF ', 6 0 ρ
 GOOD
 [20] →('*=1↑P)/POI
 [21] P← 'ALL'
 [22] POI:GS,(TKX 1 2 ρ 0 500),US,(18↑ 'QF CHECKED '),P

▽

▽ Z←RDFRM

[1] Z←(SP-1)↑,EP
 [2] Z←((ρZ)ρ(1,(SKIP-1)ρ0))/Z

▽

```

    ▽ Z←RDREC;X
[1]  Z←0
[2]  RD:X←LEGGI
[3]  →(0≠1↑FC)/0
[4]  FC←FC[2]+SKIP-1
[5]  FRAME←FRAME,[1],CONVERT X
[6]  →(250≤1↑ρFRAME)/WNG
[7]  →(EP≥FC[2])/RD
[8]  →(0<1↑ρFRAME)/0
[9]  'NO FRAMES HAVE BEEN ALLOCATED'
[10] →0
[11] WNG: 'MAX NUMBER OF FRAMES REACHED'
[12] FRAME← 250 80 ↑FRAME

```

▽

```

    ▽ Z←RDTIME;X
[1]  Z←0
[2]  RD:X←LEGGI
[3]  →(0≠1↑FC)/0
[4]  →((SKIP+1↑, 1 80 ↑FRAME)>1↑X←,CONVERT X)/RD
[5]  →(250≤1↑ρFRAME←FRAME,[1]X)/WNG
[6]  →(∧/EP>X[4+5×16])/RD
[7]  FRAME← 1 0 ↑FRAME
[8]  →(0<1↑ρFRAME)/0
[9]  'NO FRAMES HAVE BEEN ALLOCATED'
[10] →0
[11] WNG: 'MAX NUMBER OF FRAMES REACHED'
[12] FRAME← 250 80 ↑FRAME

```

▽

```

    ▽ Z←READ P;COM;PARM;SP;EP;EN;X;C;SKIP
[1]  →(0=ρ2 [SVQ 110])/ER1
[2]  COM← 4 9 ρ ' FROMRECA TORECAEVERYRECAADD
[3]  PARM← 4 1 ρ '*
[4]  →(∧/ '***' =3↑P)/EX
[5]  →(0=ρρZ←CKREAD P)/0
[6]  →(( 3 0 ↑Z)∧.= ' ')/TUT
[7]  COM← 4 9 ↑Z
[8]  PARM← 0 9 ↑Z
[9]  →EX
[10] TUT:COM←( 3 9 ↑Z),[1] 1 9 ↑COM
[11] PARM←((3,C)↑Z),[1](1,C←(1↑ρPARM)[1↑ρZ]↑Z← 0 9 ↑Z
[12] EX:→(0≠2(, 1 9 ↑COM),' ', 'PARM[1;]' )/Z←0
[13] PARM← 1 0 ↑PARM
[14] →(0≠1↑ρCOM← 1 0 ↑COM)/EX
[15] →(EP≤SP)/ERR
[16] →EN/ET
[17] →RDREC
[18] ET:→RDTIME
[19] ERR: 'START ' ,(( 2 4 ρ 'REC TIME' ) [1+EN;]), ' ≥

```

THAN END ' ,(2 4 p 'REC TIME') [1+EN;]
[20] →0
[21] ER1: 'NO SHARE ACTIVE'
[22] Z←0

▽

▽ Z←REENTER
[1] !-REENTER!
[2] Z←PACK □

▽

▽ Z←REENTERFR
[1] Z←REENTER, 'T'

▽

▽ Z←REP X
[1] FRAME← 0 80 p0
[2] Z←0

▽

▽ Z←RESET W
[1] Z←1
[2] →(Λ/ '***' ≠3↑W)/0
[3] FC←1
[4] FRAME← 0 80 p0
[5] Z←0

▽

▽ Z←RETRACT W
[1] QUESTA FUNZIONE RITRAE DALLO SHARING LE VARIABILI
OFFERTE DALLA
[2] FUNZIONE 'SHARE'
[3] W←[SVP 'FR'
[4] W←[SVP 'FC'
[5] Z←0

▽

▽ Z←SAVE X

```
[1] AQUESTA FUNZIONE SALVA LO WS COL SUO NOME CON LO STATE  
INDICATOR  
[2] ASOSPESO SULL'ISTRUZIONE DI LABEL -RESUME-  
[3] ALKUNICO ARGOMENTO ACCETTATO E' '***'  
[4] →(Λ/ '***' =3↑X)/SET  
[5] Z←1  
[6] →0  
[7] SET:SV101← 'HT'  
[8] SV101← '→RESUME'  
[9] SV101← '!)SAVE '  
[10] SASAVE←RESUME  
[11] RESUME:SV101← 'RT'  
[12] Z←0
```

▽

▽ SHAR

```
[1] AQUESTA FUNZIONE METTE IN SHARING CON I PROCESSOR 100  
[2] A101 E 110 LE VARIABILI FC,FR,SV100,SV101 E VERIFICA LA  
[3] A CORRETTEZZA DELL'OPERAZIONE  
[4] SV101← 'CMS (APL BEG'  
[5] →(2≠101 □SVO 'SV101' )/SER  
[6] →(2≠100 □SVO 'SV100' )/SER  
[7] →0  
[8] SER: 'SHARE ERROR'  
[9] OUT 'APL'
```

▽

▽ Z←SHARE X

```
[1] →(Λ/ '***' ≠3↑X)/SET  
[2] FR← 'TLMTRY FILE B1 (BIT'  
[3] FC← 'TLMTRY FILE B1 (CTL'  
[4] →SH  
[5] SET:FR←X, ' (BIT'  
[6] FC←X, ' (CTL'  
[7] SH:→(√/2≠110 □SVO 2 2 ρ 'FRFC' )/SERR  
[8] →(√/0≠FC,1↑FR)/OER  
[9] →(Λ/1=1↓3↑FC)/EER  
[10] →Z←0  
[11] SERR: 'SHARE ERROR'  
[12] Z←RETRACT '***'  
[13] →0  
[14] OER: 'OPEN ERROR'  
[15] Z←RETRACT '***'  
[16] →0  
[17] EER: 'EMPTY FILE'  
[18] Z←RETRACT '***'
```

▽


```

▽ Z←SPF C;I;Y;X;SPR;XSC;YSC;XSTR;YSTR;RIG;TIM
[1] RIG←,1
[2] →(Z←'∗'=1↑C)/0
[3] →(Z+2<I←'AB' 11↑C)/0
[4] →( 1 2 =I)/A,B
[5] A:SPR←,FRAME[; 34 80 ]:=65536
[6] →PL
[7] B:SPR←,FRAME[; 35 79 ]:=65536
[8] PL:SPR[I←(0≠SPR)/1ρSPR]←60≠SPR[(0≠SPR)/1ρSPR]
[9] →(0≠Z←0 PLT(1,(ρSPR))ρSPR)/0
[10] YSTR←∇((ρ,YSTR),1)ρ,YSTR
[11] TIM← 3 0 ∇∇ 1 0 + 24 60 60 1000 τ(,FRAME[; 31 76 ])[
XSTR]
[12] TIM[; 4 7 ]←((1↑ρTIM),2)ρ'.
[13] ASSI 500 700 500 250 1000 250
[14] OUTV TKX X,[1.1]Y[1;]
[15] GS,(TKX 1 2 ρ 400 650 ),US, 'SPIN'
[16] GS,(TKX 1 2 ρ 400 635 ),US, 'RATE'
[17] GS,(TKX 1 2 ρ 700 180 ),US, 'COUNTS'
[18] GS,(TKX 1 2 ρ 700 160 ),US, 'TIMES (HH.MM.SS)'.
[19] GS,(TKX 1 2 ρ 0 650 ),US, 'PACKAGE ' ,1↑C
[20] GS,(TKX 1 2 ρ 0 600 ),US, 'MAX VALUE FOR SPIN RATE' ,(
10 2 ∇[ /SPR), ' RPM'
[21] GS,(TKX 1 2 ρ 0 550 ),US, 'MIN VALUE FOR SPIN RATE' ,(
10 2 ∇[ /SPR), ' RPM'
[22] GS,(TKX 1 2 ρ 0 500 ),US, 'MEAN VALUE FOR SPIN RATE' ,(
9 2 ∇(+ /SPR[I])÷ρI), ' RPM'

```

▽

```

▽ START;X;Y;COM;ARG;Z;FC;FR;SV101;SV100
[1] SHAR
[2] LOOP: '...'
[3] Z←PACK [ ]
[4] COM←( 1+X←(Z=' ') 11)↑Z
[5] →(CTC<ρCOM)/ERR
[6] →(RTC<(Y←TABCOM∧.=CTC↑COM) 11)/ERR
[7] →(0≠ρ,ARG←X+Z)/EXEC
[8] ARC← '***'
[9] EXEC:Z←∇COM, ' ', 'ARG'
[10] →(∇/ 0 2 =Z)/LOOP
[11] ERR: 'INCORRECT COMMAND'
[12] →LOOP

```

▽

```

▽ Z←STATUS W;X;Y
[1] Z←1
[2] →(∧/ '***' ≠3↑W)/0
[3] Z←0
[4] →(0=1↑ρFRAME)/NO
[5] 'NUMBER OF FRAMES IN CORE: ' ,∇X←1↑ρFRAME

```

```

[6] Y← 12 3 ρ 'GENFEBMARAPR MAYJUNJULAUGSEPOCTNOVDEC'
[7] X←(1↑ 365 24 60 60 1000 τFRAME[1;1])- 0 31 59 90 120
    151 181 212 243 273 304 334 365
[8] W←(X≤0)∩1
[9] 'FROM TIME (HHMMSS) : ' ,( 3 0 ▽1↑ 24 60 60
    1000 τFRAME[1;1]), ' - ' ,( 3 0 ▽1+X[▽1+W]), ' ',Y[W-1;
    ]
[10] X←(1↑, 365 24 60 60 1000 τFRAME[1↑ρFRAME;1])- 0 31 59
    90 120 151 181 212 243 273 304 334 365.
[11] W←(X≤0)∩1
[12] 'TO TIME (HHMMSS) : ' ,( 3 0 ▽1↑, 24 60 60
    1000 τFRAME[1↑ρFRAME;1]), ' - ' ,( 3 0 ▽1+X[▽1+W]), ' '
    ,Y[W-1;]
[13] →0
[14] NO: 'NO FRAMES IN CORE'
[15] Z←2

```

▽

▽ Z←SVB C;I;SPR;Y;X;XSTR;YSTR;XSC;YSC;FUN;RIG;TIM

```

[1] RIG←,1
[2] →(0≠Z←CAL SUN C)/0
[3] →(0≠Z←0 PLT(1,ρFUN)ρFUN←90-FUN)/0
[4] YSTR←▽((ρ,YSTR),1)ρ,YSTR
[5] TIM← 3 0 ▽0 ▽1 0 ↓ 24 60 60 1000 τ(,FRAME[; 11 56 ])[
    XSTR]
[6] TIM[; 4 7 ]←((1↑ρTIM),2)ρ'. '
[7] ASSI 500 700 500 250 1000 250
[8] OUTV TKX X,[1.1]Y[1;]
[9] GS,(TKX 1 2 ρ 400 650 ),US, 'SUN'
[10] GS,(TKX 1 2 ρ 400 635 ),US, 'ANGLE'
[11] GS,(TKX 1 2 ρ 750 180 ),US, 'COUNTS'
[12] GS,(TKX 1 2 ρ 750 160 ),US, 'TIMES (HH.MM.SS)'
[13] GS,(TKX 1 2 ρ 0 600 ),US, 'PACKAGE ' ,C
[14] GS,(TKX 1 2 ρ 0 550 ),US, 'SVB RANGE' ,(▽ 10 2 ▽(L/
    FUN),[ /FUN), ' DEG'

```

▽

▽ Z←TIME X;SP;EP;SKIP;PARM;COM;RIG;EN

```

[1] Z←0
[2] COM← 3 9 ρ ' FROMRECA TORECAEVERYPECA'
[3] PARM← 3 2 ρ '*T'
[4] →('*'=1↑X)/ESEGUI
[5] →(0=ρρZ←CKREAD X)/0
[6] COM← 3 9 ↑Z
[7] PARM← ▽1 0 ↓Z←( 0 9 ↓Z),4ρ 'TT '
[8] ESEGUI:→(0≠±(, 1 9 ↑COM),' ', 'PARM[1;]' )/ERR
[9] PARM← 1 0 ↓PARM
[10] →(0≠1↑ρCOM← 1 0 ↓COM)/ESEGUI
[11] →(Z←SP≥EP)/0
[12] RIG←(SP-1)↓∩EP

```

```

[13] RIG←((ρRIG)ρ(1,(SKIP-1)ρ0))/RIG
[14] Z←TIMEPLT RIG
[15] →0
[16] ERR:Z←1

```

▽

```

▽ Z←TIMEPLT RIG;T;ΔT;X;Y;YSTR;YSC;XSTR;XSC;TIM
[1] T←(+/FRAME[RIG;4+5×18])÷8
[2] ΔT←|(-1+(1φT)-T)
[3] →(0≠Z←0 PLT(1,ρΔT)ρΔT)/0
[4] YSTR← 3 0 φ 24 60 60 1000 τYSTR
[5] YSTR←(((1+ρYSTR),9)+YSTR),(((1+ρYSTR),1)ρ'.'),((1+ρ
YSTR),3)+YSTR
[6] TIM← 3 0 φ 1 0 + 24 60 60 1000 τ,FRAME[XSTR;1]
[7] TIM[; 4 7 ]←((1+ρTIM),2)ρ'. '
[8] ASSI 500 700 500 250 1000 250
[9] OUTV TKX X,[1.1]Y[1;]
[10] ΔFSTAT
[11] →0

```

▽

```

▽ CS←TKX MXY
[1] CS← ' 1235768TXNUEDKC←M.V]RIAJG×FP[φ,' [1+, 3 1 2 φφ
32 32 τMXY]

```

▽

▽ Z←TORECA X
 [1] EN←Z←0
 [2] A:→('*'≠1↑X)/C
 [3] →('T'=-1↑X)/T1
 [4] B:EP←-1↑3↑FC
 [5] →0
 [6] C:→('T'=-1↑X)/T2
 [7] →(¬X∈ ' 0123456789')/ERR
 [8] EP←X
 [9] →(EP≤FC[3])/0
 [10] 'END RECORD NUMBER SET TO BOTTOM OF TLMTRY FILE'
 [11] →B
 [12] ERR: 'INVALID END RECORD NUMBER'
 [13] →(0≠ρ,X←REENTER)/A
 [14] Z←1
 [15] →0
 [16] T1:EP←1↑ρFRAME
 [17] →0
 [18] T2:→(¬X/(1↑X)∈ ' 0123456789')/ERT
 [19] EP←-1↑X
 [20] →(EP≤1↑ρFRAME)/0
 [21] 'END RECORD NUMBER SET TO BOTTOM OF FRAMES'
 [22] →T1
 [23] ERT: 'INVALID END RECORD NUMBER'
 [24] →(0≠ρ,X←REENTER)/A
 [25] Z←1

▽

▽ Z←TOTIME X
 [1] EN←1
 [2] A:→('*'≠1↑X)/C
 [3] EP←31622400000
 [4] →Z←0
 [5] C:→(0≠EP←CKIT X)/Z←0
 [6] 'INVALID END TIME'
 [7] →(0≠ρ,X←REENTER)/A
 [8] Z←1

▽

▽ Z←WIDTHS C;CN
 [1] CN←1
 [2] →(0≠Z←CN WINF C)/0
 [3] GS,(TKX 1 2 ρ 350 650),US, 'WIDTHS'
 [4] GS,(TKX 1 2 ρ 350 635),US, '(DEG)'
 [5] GS,(TKX 1 2 ρ 700 180),US, 'COUNTS'
 [6] GS,(TKX 1 2 ρ 700 160),US, 'TIMES'

▽

▽ Z←CN WINF C;I;EIN;EOUT;X;Y;XSC;YSC;XSTR;YSTR;TIM;RIG
 [1] RIG←,1
 [2] →(0≠Z←CN CALINF C)/0
 [3] LL:→(0≠Z←0 PLT EOUT,[0.1]EIN)/0
 [4] YSTR←▽((ρ,YSTR),1)ρ,YSTP
 [5] TIM← 3 0 ▽ 1 0 + 24 60 60 1000 τ,FRAME[; 1 1]
 [6] TIM[; 4 7]←((1+ρTIM),2)ρ'.
 [7] TIM←TIM[XSTR;]
 [8] ASSI 500 700 500 250 1000 250
 [9] LOOP:→(0=1+ρY)/ST
 [10] OUTV TKX X,[1.1]Y[1;]
 [11] Y← 1 0 +Y
 [12] →LOOP
 [13] ST:GS,(TKX 1 2 ρ 0 500),US, 'PACKAGE ' ,C
 [14] GS,(TKX 1 2 ρ 0 450),US, 'INF1 RANGE ' ,(▽ 10 2 ▽(L/
 EIN),[/EIN), ' DEG'
 [15] GS,(TKX 1 2 ρ 0 400),US, 'INF2 RANGE ' ,(▽ 10 2 ▽(L/
 EOUT),[/EOUT), ' DEG'

▽

▽ ΔTSTAT;ΔTM;SG
 [1] GS,(TKX 1 2 ρ 725 185),US, 'FRAMES'
 [2] GS,(TKX 1 2 ρ 725 170),US, 'TIMES (HH.MM.SS)'
 [3] GS,(TKX 1 2 ρ 320 650),US, 'ΔTIME'
 [4] GS,(TKX 1 2 ρ 286 630),US, '(HH.MM.SS.MS)'
 [5] GS,(TKX 1 2 ρ 0 550),US, 'NUM OF MF' , 21 0 ▽ρRIG
 [6] GS,(TKX 1 2 ρ 165 470),US, 'DD HH MM SS'
 [7] GS,(TKX 1 2 ρ 0 450),US, 'MAX VALUE FOR ΔT ' ,(4 0
 ▽ 1+ 0 24 60 60 1000 τ[/ΔT)
 [8] GS,(TKX 1 2 ρ 0 400),US, 'MIN VALUE FOR ΔT ' ,(4 0
 ▽ 1+ 0 24 60 60 1000 τ[/ΔT)
 [9] GS,(TKX 1 2 ρ 0 350),US, 'MEAN VALUE FOR ΔT ' ,(4 0
 ▽ 1+ , 0 24 60 60 1000 τΔTM←[0.5+(+/ΔT)÷ρΔT)

▽

APPENDICE B

Cross reference dello spazio di lavoro

STATS
THERE ARE 220 NAMES
733 CHARACTERS IN THE NAMES
60 FUNCTIONS, 829 LINES OF CODE
13.81666667 LINES/FUNCTION AVERAGE
14 GLOBAL VARIABLE NAMES
23 SEMI-GLOBAL VARIABLE NAMES
52 LOCAL VARIABLE NAMES
78 LABEL NAMES
AND 0 UNREFERENCED VARIABLES
PRIME FUNCTIONS: ADD CKENDRD CMS CP END EVERYRECA EVERYTIME FLAG
FROMRECA FROMTIME HELP ILF INF1 INF2 PLOT QF RDERM
READ REP RESET SHARE SPF START STATUS SVB TIME TORECA
TOTIME WIDTHS

FUNCTIONS

60 9

FUNCTIONS

ADD
ASSI
BEAM
CALINF
CALQF
CALSUN
CBEAM
CKENDRD
CKIT
CKOF
CKREAD
CMS
COMDES
COMDS
CONV
CONVERT
CP
CPCMS
END
EVERYRECA
EVERYTIME
FLAG
FROMRECA
FROMTIME
HELP
ILF
INF
INF1
INF2
LEGGI
OUT
OUTV
PACK
PLOT
PLT
QF
RDFRM
RDREC
RDTIME
READ
REENTER
REENTERFF
REP
RESET
RETRACT
SAVE
SHAR
SHARE
SPF
START
STATUS
SVB
TIME
TIMEPLT
TKX
TORECA
TOTIME
WIDTHS
WINF
ATSTAT

ALL LISTREF ALL
GLOBAL VARIABLES:

CTC: START
C: FLAG
EIN[FLAG]: CALINF
EIN[INF]: CALINF
EIN[WINF]: CALINF
EN: CKENDRD EVERYRECA EVERYTIME FROMRECA FROMTIME TORECA TOTIME
EOUT[FLAG]: CALINF
EOUT[INF]: CALINF
EOUT[WINF]: CALINF
EP[READ]: RDREC RDTIME
EP: CKENDRD RDFRM TORECA TOTIME
FC: CKENDRD FROMRECA FROMTIME LEGGI RDREC RDTIME RESET SHAPE
TORECA
FRAME: CALINF CALSUN CKENDRD FLAG FROMRECA ILF INF PLOT QF RDREC
RDTIME REP RESET SPF STATUS SVB TIMEPLT TORECA WINF
FR: LEGGI SHARE
FUN[FLAG]: CALSUN
FUN[SVB]: CALSUN
GS: ASSI ILF INF1 INF2 INF OUTV QF SPF SVB WIDTHS WINF Δ TSTAT
RDSUB: CKREAD
RIG[ILF]: PLT
RIG[INF]: PLT
RIG[QF]: PLT
RIG[SPF]: PLT
RIG[SVB]: PLT
RIG[TIMEPLT]: PLT Δ TSTAT
RIG[WINF]: PLT
RTC: START
SENS: CKQF
SKIP[READ]: RDREC RDTIME
SKIP: EVERYRECA EVERYTIME RDFRM
SP: FROMRECA FROMTIME RDFRM
SV100: CMS CP
SV101[START]: OUT SHAR
SV101: SAVE
S Δ OUT: OUT
S Δ SAVE: SAVE
TABCOM: START
TIM[ILF]: ASSI
TIM[INF]: ASSI
TIM[QF]: ASSI
TIM[SPF]: ASSI
TIM[SVB]: ASSI
TIM[TIMEPLT]: ASSI
TIM[WINF]: ASSI
US: ASSI ILF INF1 INF2 INF OUTV QF SPF SVB WIDTHS WINF Δ TSTAT
XSC[ILF]: ASSI PLT
XSC[INF]: ASSI PLT

XSC[QF]: ASSI PLT
XSC[SPF]: ASSI PLT
XSC[SVB]: ASSI PLT
XSC[TIMEPLT]: ASSI PLT
XSC[WINF]: ASSI PLT
XSTR[ILF]: ASSI PLT
XSTR[INF]: ASSI PLT
XSTR[QF]: ASSI PLT
XSTR[SPF]: ASSI PLT
XSTR[SVB]: ASSI PLT
XSTR[TIMEPLT]: ASSI PLT
XSTR[WINF]: ASSI PLT
X[FLAG]: CALINF
X[ILF]: PLT
X[INF]: CALINF PLT
X[QF]: PLT
X[SPF]: PLT
X[SVB]: PLT
X[TIMEPLT]: PLT
X[WINF]: CALINF PLT
YSC[ILF]: ASSI PLT
YSC[INF]: ASSI PLT
YSC[QF]: ASSI PLT
YSC[SPF]: ASSI PLT
YSC[SVB]: ASSI PLT
YSC[TIMEPLT]: ASSI PLT
YSC[WINF]: ASSI PLT
YSTR[ILF]: ASSI PLT
YSTR[INF]: ASSI PLT
YSTR[QF]: ASSI PLT
YSTR[SPF]: ASSI PLT
YSTR[SVB]: ASSI PLT
YSTR[TIMEPLT]: ASSI PLT
YSTR[WINF]: ASSI PLT
Y[ILF]: PLT
Y[INF]: PLT
Y[QF]: PLT
Y[SPF]: PLT
Y[SVB]: PLT
Y[TIMEPLT]: PLT
Y[WINF]: PLT
 ΔT [TIMEPLT]: ΔT STAT
[AV: CBEAM
[WC: PLOT
[SVO: SHARE SHAR
[SVO: READ
[SVR: RETRACT
[]: CBEAM REENTER START

FUNCTIONS:

Z←ADD X
(2 LINES)

ASSI CART ;PA;XX;YY
FUNS REFD: OUTV TKX
VARS REFD: GS TIM US XSC XSTR YSC YSTR
CALLED BY: ILF INF QF SPF SVB TIMEPLT WINF
(16 LINES)

Z←BEAM ;Z1
FUNS REFD: CBEAM
CALLED BY: PLOT
(3 LINES)

Z←CN CALINF C ;EW;I;SPR
VARS REFD: EIN EOUT FRAME X
CALLED BY: FLAG INF WINF
(21 LINES)

Z←CALQF A ;PARM
CALLED BY: FLAG QF
(5 LINES)

Z←CALSUN C ;I;SPR
VARS REFD: FRAME FUN
CALLED BY: FLAG SVB
(13 LINES)

Z←CBEAM
VARS REFD: [AV]
CALLED BY: BEAM
(4 LINES)

Z←CKENDRD T
FUNS REFD: CONV
VARS REFD: EN EP FC FRAME
(11 LINES)

Z←CKIT T ;T1;X
FUNS REFD: PACK
CALLED BY: FROMTIME TOTIME
(17 LINES)

Z*CKQF P ;J;PARM;PR;TYP;V;X
VARS REFD: SEMS
CALLED BY: FLAG QF
(13 LINES)

Z*CKREAD P ;C;I;OP;PR1;PR2;PR3;TCOM;TPARM;X
FUNS REFD: PACK
VARS REFD: RDSUB
CALLED BY: READ TIME
(43 LINES)

Z*CMS A
FUNS REFD: CPCMS
VARS REFD: SV100
(9 LINES)

COMDES
CALLED BY: COMDS
(32 LINES)

Z*COMDS A
FUNS REFD: COMDES
CALLED BY: HELP
(7 LINES)

Z*CONVERT T
FUNS REFD: CONV
CALLED BY: RDREC RDTIME
(2 LINES)

Z*I CONV T
CALLED BY: CKENDRD CONVERT FROMTIME
(2 LINES)

Z*CPCMS A
FUNS REFD: SAVE
CALLED BY: CMS CP
(13 LINES)

Z*CP A
FUNS REFD: CPCMS
VARS REFD: SV100
(8 LINES)

Z←END A

FUNS REFD: OUT SAVE
(6 LINES)

Z←EVERYRECA X

FUNS REFD: REENTER
VARS REFD: EN SKIP
(11 LINES)

Z←EVERYTIME X

FUNS REFD: PACK REENTER
VARS REFD: EN SKIP
(11 LINES)

Z←FLAG AA ;DIM;EIN;EOUT;FUN;I;OP;PARM;SEL;SPR;TAB;TIB;TLR;X;ZZ
FUNS REFD: CALINF CALQF CALSUM CKQF PACK REENTER
VARS REFD: C FRAME
(64 LINES)

Z←FROMRECA X

FUNS REFD: REENTERER REENTER
VARS REFD: EN FC FRAME SP
(23 LINES)

Z←FROMTIME X

FUNS REFD: CKIT CONV LEGGI REENTER
VARS REFD: EN FC SP
(22 LINES)

Z←HELP H ;CM;TT1;TT2;TT;YY

FUNS REFD: COMDS
(38 LINES)

Z←ILF C ;IN;I;OUT;RIG;SPR;TIM;XSC;XSTR;X;YSC;YSTR;YY;Y

FUNS REFD: ASSI OUTV PLT TKX
VARS REFD: FRAME GS US
(28 LINES)

Z←INF1 C ;CN

FUNS REFD: INF TKX
VARS REFD: GS US
(7 LINES)

Z*INF2 C ;CN
FUNS REFD: INF TKX
VARS REFD: GS US
(7 LINES)

Z*CN INF C ;EIN;EOUT;I;RIG;TIM;XSC;XSTR;X;YSC;YSTR;Y
FUNS REFD: ASSI CALINF OUTV PLT TKX
VARS REFD: FRAME GS US
CALLED BY: INF1 INF2
(16 LINES)

Z*LEGGI
VARS REFD: FC FR
CALLED BY: FROMTIME RDREC RDTIME
(4 LINES)

OUTV A ;Z1;Z2;PW
VARS REFD: GS US
CALLED BY: ASSI ILF INF QF SPF SVB TIMEPLT WINF
(9 LINES)

Z*OUT X
VARS REFD: SV101 SΔOUT
CALLED BY: END SHAR
(10 LINES)

Z*PACK S ;X
CALLED BY: CKIT CKREAD EVERYTIME FLAG REENTER START
(4 LINES)

Z*PLOT OP ;ARG;A;COM;CTC;I;RTC;TBC;X
FUNS REFD: BEAM
VARS REFD: FRAME INC
(18 LINES)

Z*FLAG PLT M ;MM;ΔY
VARS REFD: RIG XSC XSTR X YSC YSTR Y
CALLED BY: ILF INF QF SPF SVB TIMEPLT WINF
(27 LINES)

Z*QF P ;GOOD;RIG;TIM;XSC;XSTR;X;YSC;YSTR;Y;ZZ
FUNS REFD: ASSI CALOF CKQF OUTV PLT TKX
VARS REFD: FRAME GS US
(23 LINES)

Z←RDPRM
VARS REFD: EP SKIP SP
(3 LINES)

Z←RDREC ;X
FUNS REFD: CONVERT LEGGI
VARS REFD: EP FC FRAME SKIP
CALLED BY: READ
(13 LINES)

Z←RDTIME ;X
FUNS REFD: CONVERT LEGGI
VARS REFD: EP FC FRAME SKIP
CALLED BY: READ
(13 LINES)

Z←READ P ;COM;C;EN;EP;PARAM;SKIP;SP;X
FUNS REFD: CKREAD RDREC RDTIME
VARS REFD: SVQ
(23 LINES)

Z←REENTERFR
FUNS REFD: REENTER
CALLED BY: FROMRECA TORECA
(2 LINES)

Z←REENTER
FUNS REFD: PACK
VARS REFD:
CALLED BY: EVERYRECA EVERYTIME FLAG FROMRECA FROMTIME
REENTERFR TORECA TOTIME
(3 LINES)

Z←REP X
VARS REFD: FRAME
(3 LINES)

Z←RESET W
VARS REFD: FC FRAME
(6 LINES)

Z←RETRACT W
VARS REFD: SVR
CALLED BY: SHARE
(6 LINES)

Z←SAVE X
VARS REFD: SV101 SASAVE
CALLED BY: CPCMS END
(13 LINES)

Z←SHARE X
FUNS REFD: RETRACT
VARS REFD: FC FR SVO
(19 LINES)

SHAR
FUNS REFD: OUT
VARS REFD: SV101 SVO
CALLED BY: START
(10 LINES)

Z←SPF C ;I;RIG;SPR;TIM;XSC;XSTR;X;YSC;YSTR;Y
FUNS REFD: ASSI OUTV PLT TKX
VARS REFD: FRAME GS US
(23 LINES)

START ;ARG;COM;FC;FR;SV100;SV101;X;Y;Z
FUNS REFD: PACK SHAR
VARS REFD: CTC RTC TABCOM
(13 LINES)

Z←STATUS W ;X;Y
VARS REFD: FRAME
(16 LINES)

Z←SVB C ;FUN;I;RIG;SPR;TIM;XSC;XSTR;X;YSC;YSTR;Y
FUNS REFD: ASSI CALSUN OUTV PLT TKX
VARS REFD: FRAME GS US
(15 LINES)

Z←TIMEPLT RIG ;TIM;T;XSC;XSTR;X;YSC;YSTR;Y;ΔT
FUNS REFD: ASSI OUTV PLT TKX ΔTSTAT
VARS REFD: FRAME
CALLED BY: TIME
(12 LINES)

Z←TIME X ;COM;EN;EP;PARM;RIG;SKIP;SP
FUNS REFD: CKREAD TIMEPLT
(17 LINES)

CS←TKX MXY

CALLED BY: ASSI IIF INF1 INF2 INF OF SPF SVB TIMEPLT WIDTHS
WINF ΔTSTAT

(2 LINES)

Z←TORECA X

FUNS REFD: REENTERFR REENTER
VARS REFD: EN EP FC FRAME

(26 LINES)

Z←TOTIME X

FUNS REFD: CKIT REENTER
VARS REFD: EN EP

(9 LINES)

Z←WIDTHS C ;CN

FUNS REFD: TKX WINF
VARS REFD: GS US

(7 LINES)

Z←CN WINF C ;EIN;EQUT;I;RIG;TIM;XSC;XSTR;X;YSC;YSTR;Y

FUNS REFD: ASSI CALINF OUTV PLT TKX
VARS REFD: FRAME GS US

CALLLED BY: WIDTHS

(16 LINES)

ΔTSTAT ;SG;ΔTM

FUNS REFD: TKX
VARS REFD: GS RIG US ΔT
CALLED BY: TIMEPLT

(10 LINES)

TOPO

PAGE 1

- *ADD
- *CKENDRD
 - CONV
- *CMS
 - CPCMS
 - | ◦SAVE
- *CP
 - CPCMS (ALREADY DETAILED)
- *END
 - OUT
 - SAVE
- *EVERYRECA
 - REENTER
 - | ◦PACK
- *EVERYTIME
 - PACK
 - REENTER (ALREADY DETAILED)
- *FLAG
 - CALINF
 - CALQF
 - CALSUN
 - CKQF
 - PACK
 - REENTER (ALREADY DETAILED)
- *FROMRECA
 - REENTERFR
 - | ◦REENTER (ALREADY DETAILED)
 - REENTER (ALREADY DETAILED)
- *FROMTIME
 - CKIT
 - | ◦PACK
 - CONV
 - LEGGI
 - REENTER (ALREADY DETAILED)
- *HELP
 - COMDS
 - | ◦COMDES
- *ILF
 - ASSI
 - | ◦OUTV
 - | ◦TKX
 - OUTV
 - PLT
 - TKX
- *INF1
 - INF
 - | ◦ASSI (ALREADY DETAILED)
 - | ◦CALINF
 - | ◦OUTV
 - | ◦PLT
 - | ◦TKX
 - TKX
- *INF2
 - INF (ALREADY DETAILED)
 - TKX
- *PLOT

1

- BEAM
- | ◦CBEAM
- *QF
 - ASSI (ALREADY DETAILED)
 - CALQF
 - CKOF
 - OUTV
 - PLT
 - TKX
- *RDFRM
- *FEAD
 - CKREAD
 - | ◦PACK
 - RDREC
 - | ◦CONVERT
 - | | ◦CONV
 - | ◦LEGGI
 - RDTIME
 - | ◦CONVERT (ALREADY DETAILED)
 - | ◦LEGGI
- *REP
- *RESET
- *SHARE
 - RETRACT
- *SPF
 - ASSI (ALREADY DETAILED)
 - OUTV
 - PLT
 - TKX
- *START
 - PACK
 - SHAR
 - | ◦OUT
- *STATUS
- *SVB
 - ASSI (ALREADY DETAILED)
 - CALSUN
 - OUTV
 - PLT
 - TKX
- *TIME
 - CKREAD (ALREADY DETAILED)
 - TIMEPLT
 - | ◦ASSI (ALREADY DETAILED)
 - | ◦OUTV
 - | ◦PLT
 - | ◦TKX
 - | ◦ΔTSTAT
 - | | ◦TKX
- *TORECA
 - REENTERFR (ALREADY DETAILED)
 - REENTER (ALREADY DETAILED)
- *TOTIME
 - CKIT (ALREADY DETAILED)
 - REENTER (ALREADY DETAILED)
- *WIDTHS
 - TKX

1

PAGE 3

◦WINF

| ◦ASSI (ALREADY DETAILED)

| ◦CALINF

| ◦OUTV

| ◦PLT

| ◦TKX

SHOWREFS

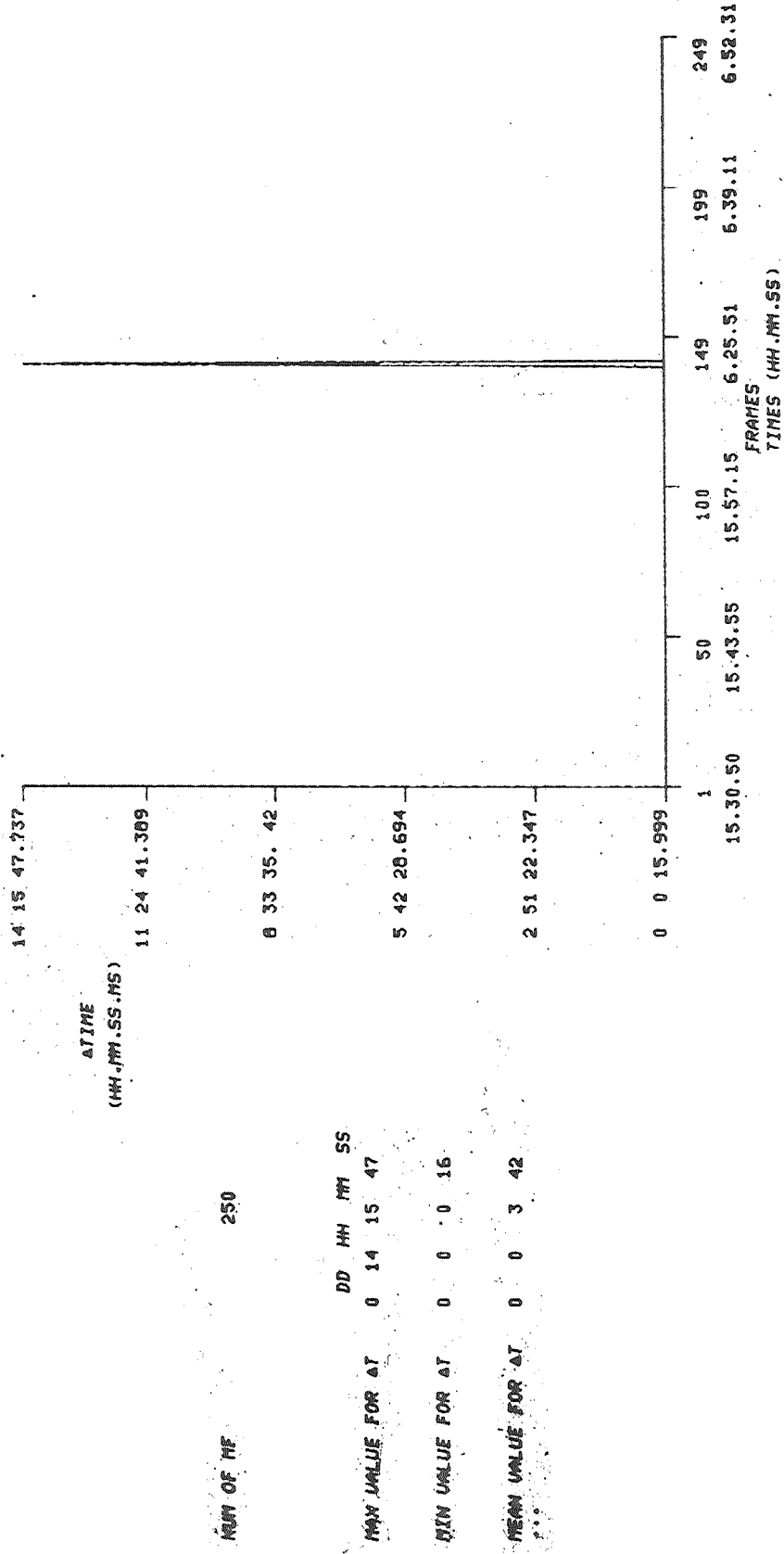
F|AABCCCCCCCCCCCCCEEEFFFHJ|IIILLOPPFPQPPRRRRPSSSSSSSTTTTWWA
 |DSEAAABKKKKMCOCCOPPNVVRPEINNNNEUALLFDDDEEEFFFAHHPTTVIIKCCIIT
 |DSALLLEEIORSMMNNC DEEACOLPFFFGTTCOT PPTAEEPSTVAFAAARMXRTDNS
 |IMIOSANTFE DDVVM RRCMP 12 GV KT| REIDNN EREER RT EE|EITFT
 |NFUMD |A ESE S YY PT| I | MCM|TT TA F TU P |CMH A
 |F|N R |D S R RT EI| | E|EE C |S L |ΔES T
 | | D | T EI CM| | |RR T | T | |
 | | | CM ΔE| | | F | | |
 | | | ΔE | | | R | | |

N

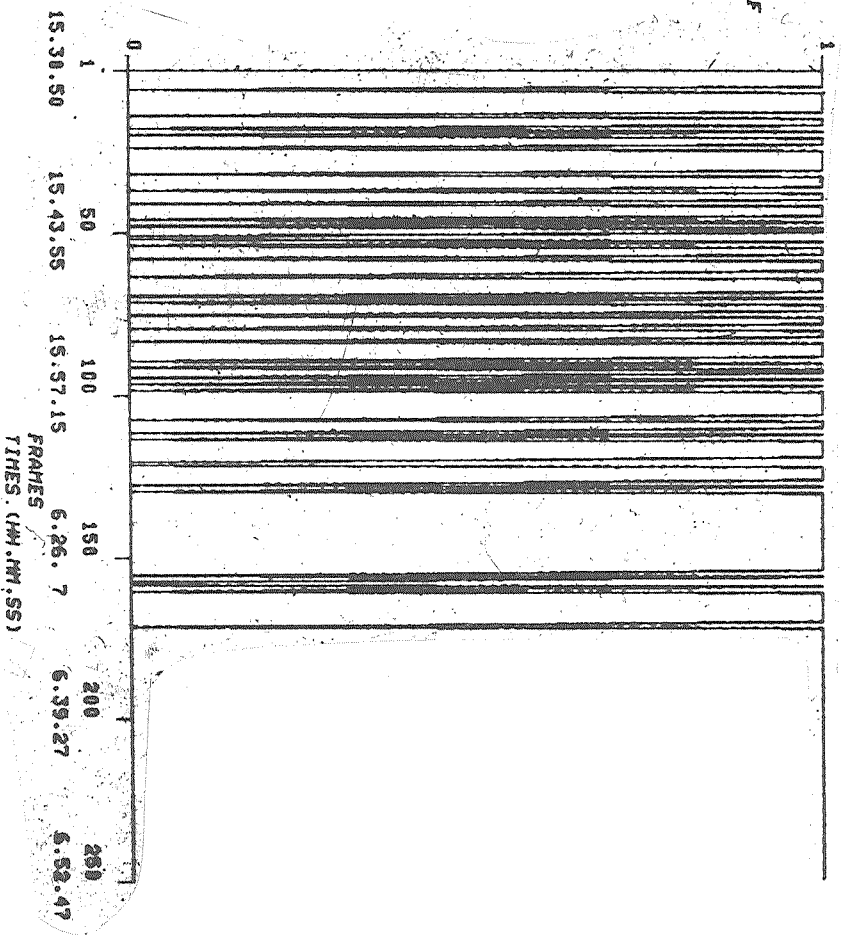
| | | | | | | | | | | | | | | | | | | |
|--------|---|---|----|-----|----|------|-----------|-------------------|-------------------|-------|-------|----|-------|-------|-------|----|------|-----|
| TT2 | | | | | | L | | | | | | | | | | | | L |
| TT | | | | | | L | | | | | | | | | | | | L |
| TUT | | | | | | T | | | | | | T | | | | | | T |
| TYP | | | | | | L | | | | | | | | | | | | L |
| TO | | | | | | +T | | | | | | | | | | | | +T |
| T1 | | | L | | | T | | | | | | | | | | | T | B |
| T2 | | | | | | T | | | | | | | | | | | T | T |
| T3 | | | | | | T | | | | | | | | | | | | T |
| T | | | LL | | LL | | | | | | | | | | | L | | L |
| UP | | | | | | +T | | | | | | | | | | | | +T |
| US | G | | | | | GGGG | G | G | | | | G | G | | | | GGGG | |
| V | | | L | | | | | | | | | | | | | | | L |
| WIDTHS | | | | | | | | | | | | | | | | | | ° |
| WID | | | | | | T | | | | | | | | | | | | T |
| WINF | | | | | | | | | | | | | | | | | | G+G |
| WNG | | | | | | T | | | | | | TT | | | | | | T |
| W | | | T | | | | | | | | | LL | L | | | | B | |
| XSC | G | | | | | L | L | GL | | | | L | LL | | | | L | S |
| XSTR | G | | | | | L | L | GL | | | | L | LL | | | | L | S |
| XX | | | L | | | | | | | | | | | | | | | L |
| X | L | G | | LLL | | LLLL | L | L | LLL | CL | L | L | LL | LLLLL | | LL | L | S |
| YSC | G | | | | | L | L | GL | | | | L | LL | | | | L | S |
| YSTR | G | | | | | L | L | GL | | | | L | LL | | | | L | S |
| YY | L | | | | | LL | | | | | | | | | | | | L |
| Y | | | | | | +L | -L | -CL | | | | | | LLLLL | | | | L+S |
| ZZ | | | | | | L | | | | | L | | | | | | | L |
| Z1 | | L | | | | | | | | | L | | | | | | | L |
| Z2 | | | | | | | | | | | L | | | | | | | L |
| Z | | | | | | L | LLLLLLLLL | LLLLLLLLLLLLLLLLL | LLLLLLLLLLLLLLLLL | LLLLL | LLLLL | | LLLLL | | LLLLL | | L | |
| ΔTM | | | | | | | | | | | | | | | | | | LL |
| ΔTSTAT | | | | | | | | | | | | | | | | G | | G |
| ΔT | | | | | | | | | | | | | | | | L | | GS |
| ΔY | | | | | | | | | | | L | | | | | | | L |
| □AV | | | G | | | | | | | | | | | | | | | C |
| □NC | | | | | | | | | | G | | | | | | | | G |
| □PW | | | | | | | | | L | | | | | | | | | L |
| □SVO | | | | | | | | | | | | | GC | | | | | C |
| □SVQ | | | | | | | | | | | G | | | | | | | G |
| □SVR | | | | | | | | | | | | | G | | | | | G |
| □ | | | G | | | | | | | | | G | | G | | | | G |

APPENDICE C

Esempio di seduta al terminale



NUMBER OF MF 250
OF CHECKED ALL



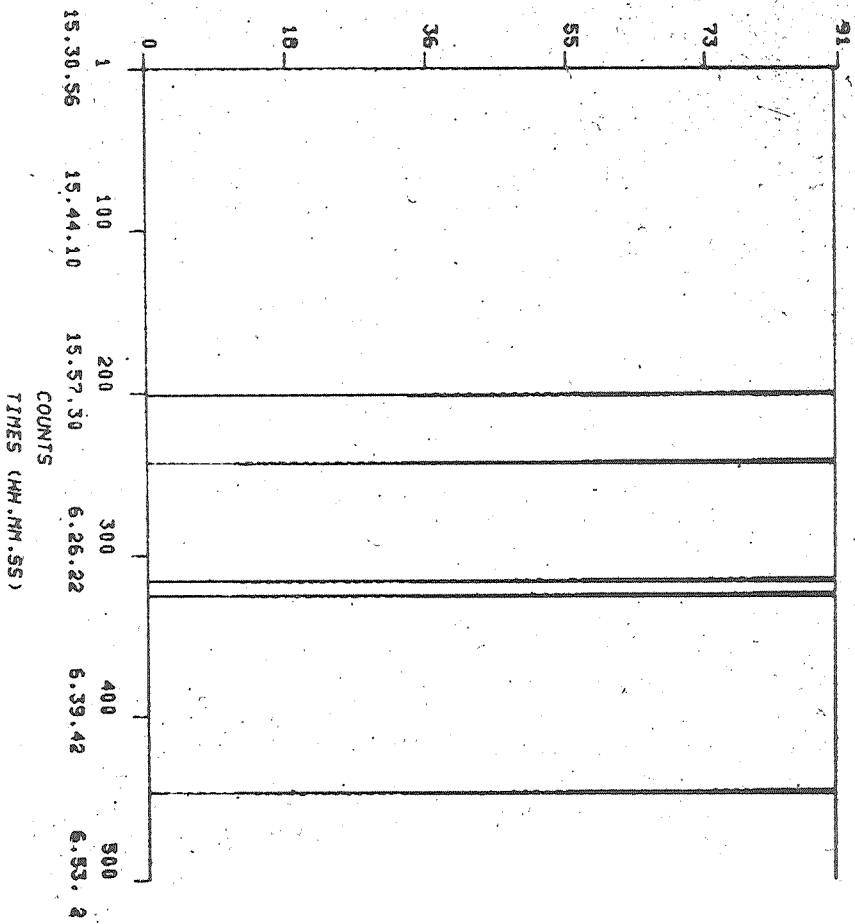
PACKAGE A

MAX VALUE FOR SPIN RATE 90.37 RPM

MIN VALUE FOR SPIN RATE .00 RPM

MEAN VALUE FOR SPIN RATE 90.36 RPM

SPIN RATE



NUMBER OF MF 249

OF CHECKED INF1 A

FLAG INF1A

INCORRECT COMMAND

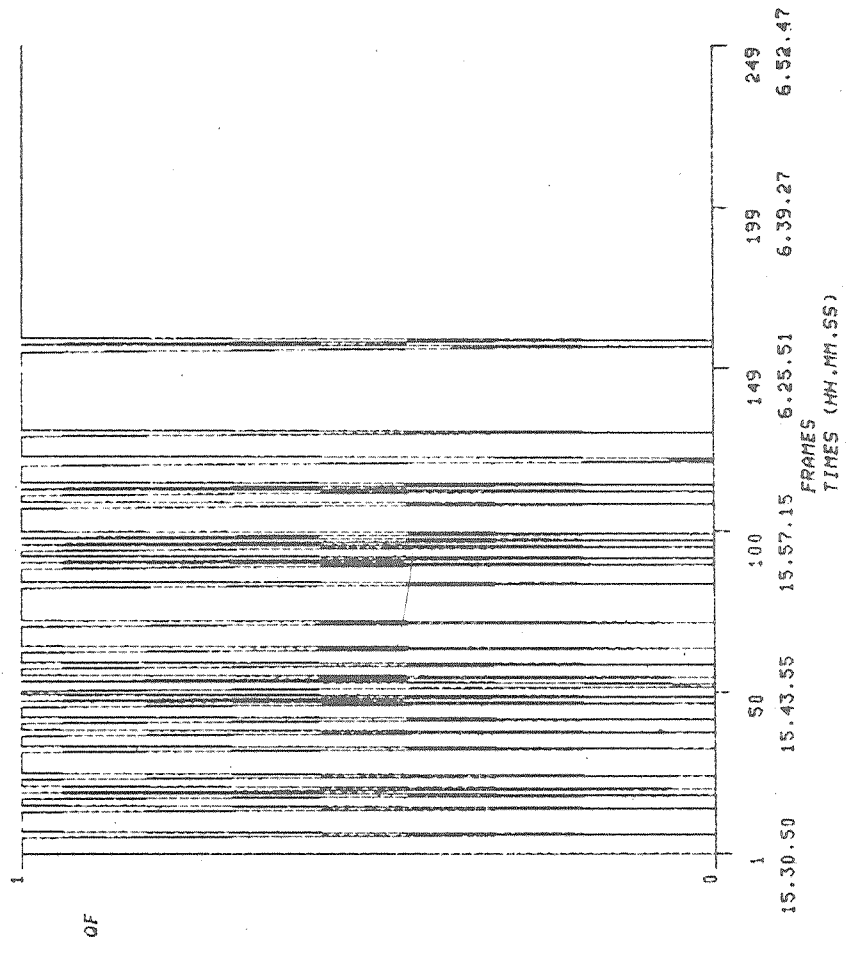
FLAG OF INF1 A

CHECKED FIELDS

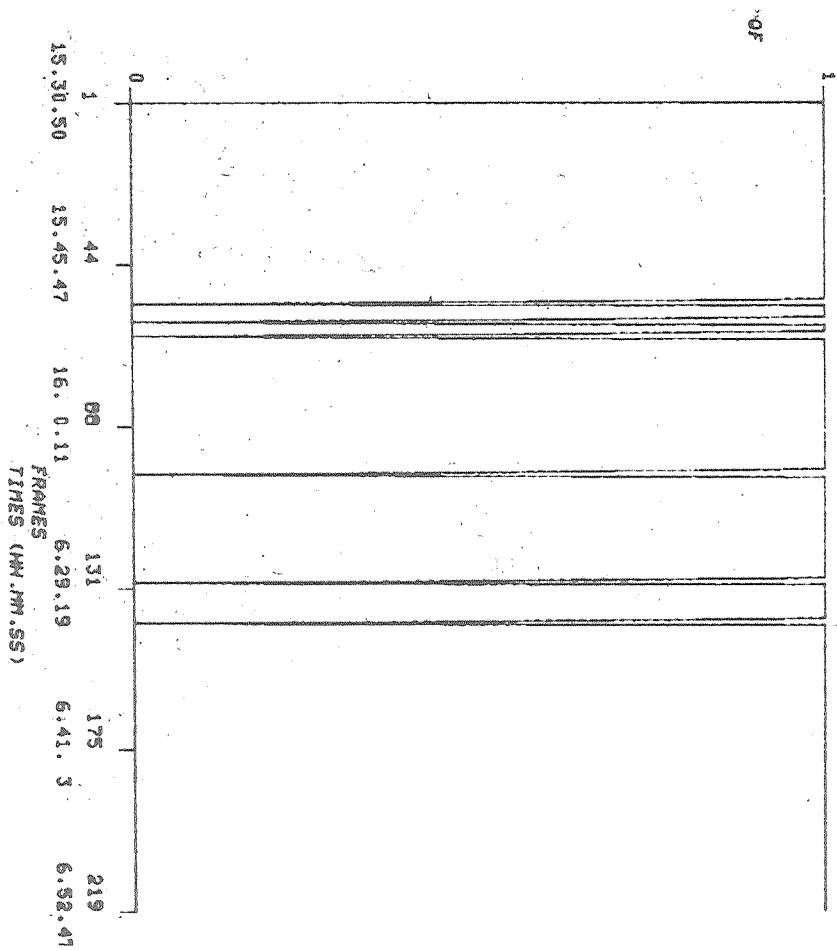
NUMBER OF FRAMES FLAGGED

...

OF INF1 A 30



NUMBER OF MF 219
OF CHECKED ALL
...

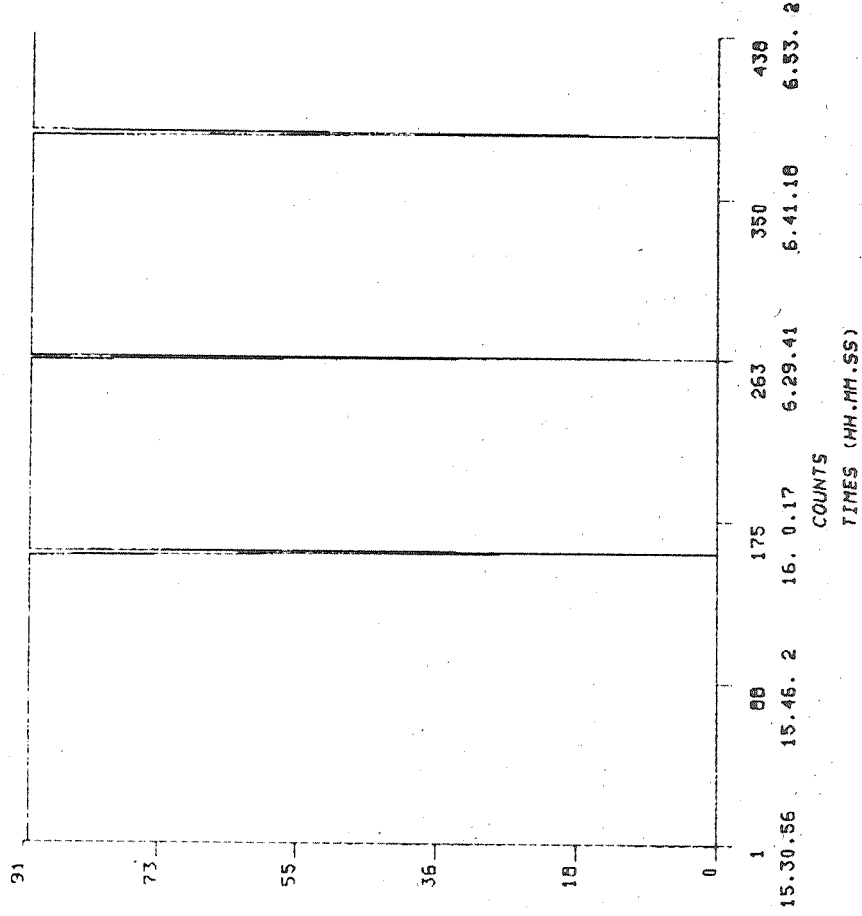


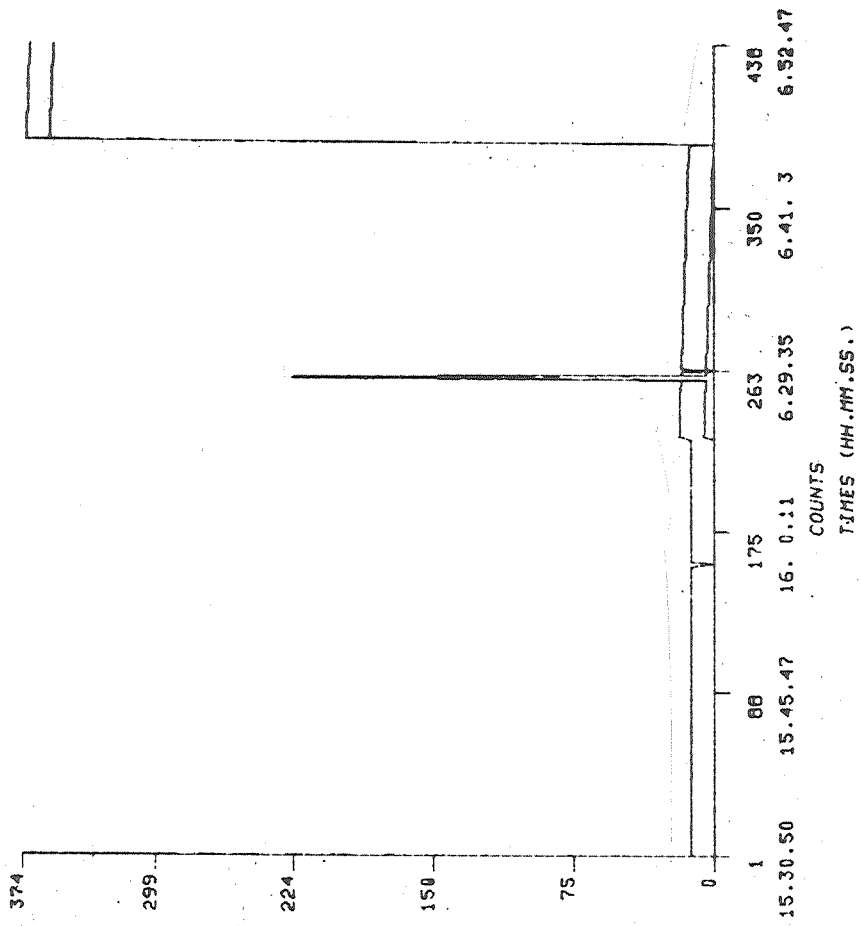
PACKAGE A

MAX VALUE FOR SPIN RATE 90.37 RPM

MIN VALUE FOR SPIN RATE .00 RPM

MEAN VALUE FOR SPIN RATE 90.36 RPM
...





INF1
(DEG)

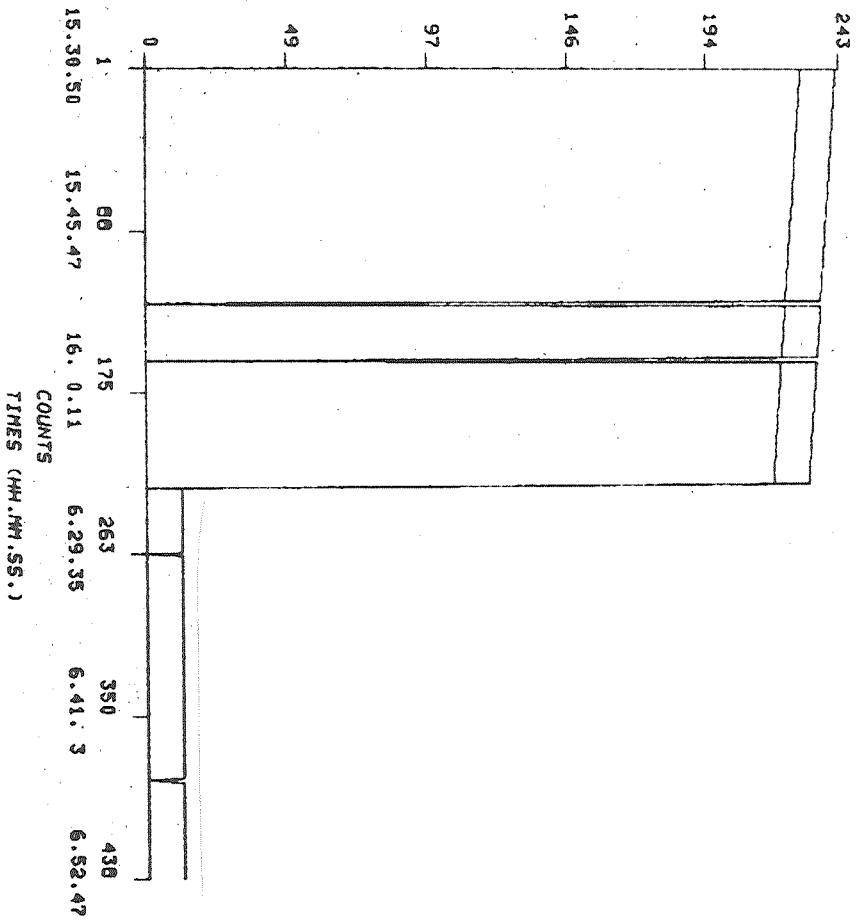
PACKAGE A

EIN RANGE .00 359.94 DEG

ECUT RANGE .00 373.26 DEG

PACKAGE A
 IN RANGE .00 228.95 DEG
 OUT RANGE .00 241.95 DEG

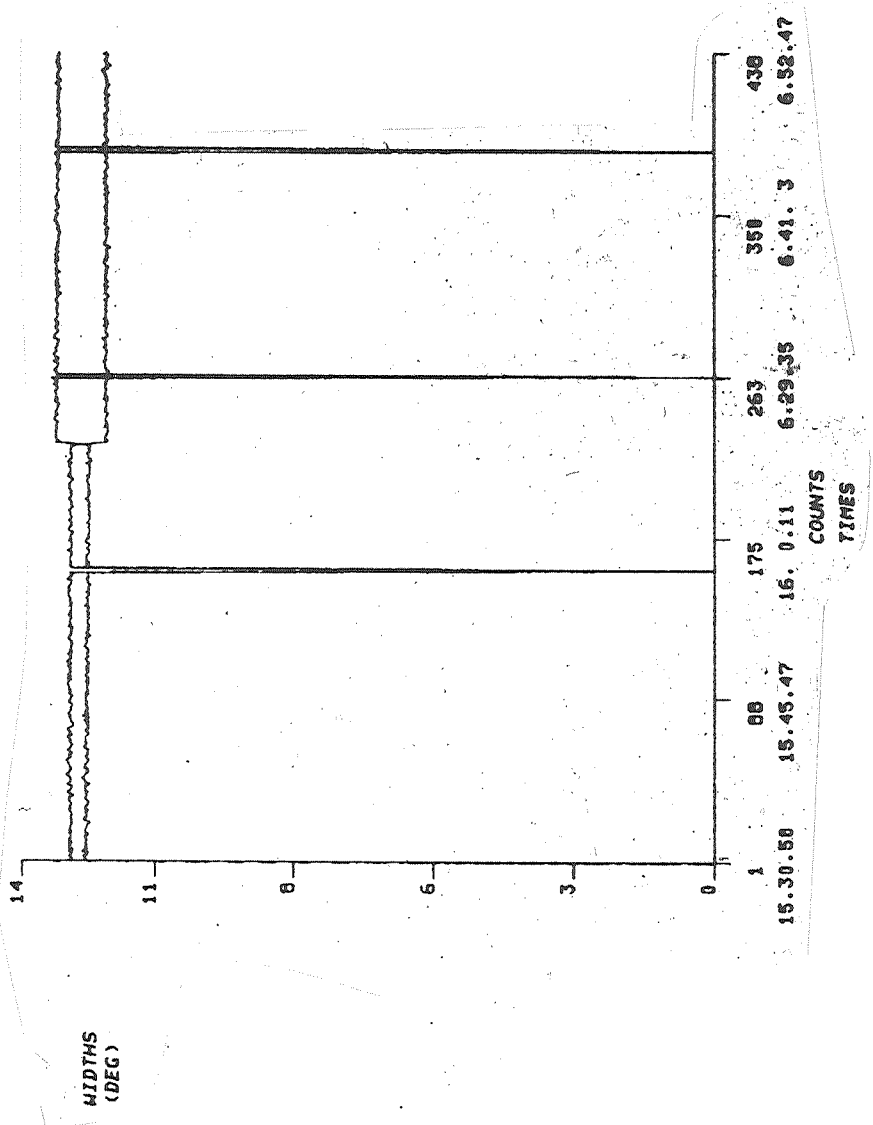
IMP2
 (DEG)



PACKAGE A

INF1 RANGE .00 13.39 DEG

INF2 RANGE .00 13.07 DEG



PACKAGE A

SUB RANGE 90.00 95.39 DEG
 STATUS
 NUMBER OF FRAMES IN CORE: 219
 FROM TIME (HHMMSS) : 15 30 50 - 3 APR
 TO TIME (HHMMSS) : 6 52 47 - 4 APR
 END NOSAVE

