ONLINE DATA-LOGGING IN GMRT

crs/July90

OBSERVING FILE

This set of files has not fully been thought about yet. Given below are some essential contents of the files, far from complete.

obsfile_N: This is the observing file for subarray_N. The first line of this file contains

obs_prog;observer;epoch;date.start;lst.start;lst.stop

where obs_prog is the name of the scheduled observation, and date.start, lst.start and lst.stop correspond to allocated schedule for the obs_prog. Subsequent lines in the file contain scan details, which will be processed only from lst.start. The lst.stop defined in the first line is an absolute limit and will override any definition contained in the scan_records to follow. The session will be terminated at lst.stop irrespective of whether the scans have been completed or not.

The operator, however, can redefine lst.stop by using a privileged command.

The scan details include sourcename, coordinates at the epoch indicated in the first line, lst_duration of observation. The lst_duration can be specified by one of the following ways:

lst.start.timed lst.stop.timed
lst.start duration
duration

When lst.start.timed lst.stop.timed are specified observations are restricted to the available interval between the two specified lst values. In particular, if the current lst exceeds (or within 1 min of) lst.stop.timed, the whole scan is skipped. When duration is specified, it will be counted from lst.start if specified or the current time. Also, no minimum time limit (like 1 min above) is imposed when duration is specified instead of lst.stop.timed.

Note that the duration will include move_time. If there was no previous activity before the scan, the program will move the telescope to the appropriate point and wait till scan is due to be scheduled. Unless the observer can anticipate this while preparing obsfile_N, proper allowance must be made for telescope movement while specifying duration or lst.stop.timed.

Cable-wrapping can be enabled, if necessary, by the observer by inserting a line containing the string

cable.wrap

alone in a line (rest of line should be blank). This line can be inserted anywhere in the obsfile_N; in the absence of this, loss of observation may result from the cable-wrap initiated by computer when the end of cable is reached in any direction.

When the desired coordinate can be achieved by movement in either possible direction, normally the shortest path will be chosen by the computer. However, one can override this by explicitly specifying direction of motion by the parameter

cable.positive or cable.negative .

MONITOR DATABASE AT THE SERVER

Summary of files:

filename	contents	related pgm
commands.log	all telescope movement commands	rd_cmd
telposN.log	position/online flags for subarray_N	rd_tel
monN.log	monitor data-base for subarray_N	rd_mon

(a) commands.log: This file will record all commands which relate to telescope movement, along with the time at which command was initiated from the online computer. Each line (variable length) will contain time-stamp, command, target position (if any), and the list of antennas to which command was sent. Any group names like subarrays are expanded into individual antenna names in this line.

A separate software ("rd_cmd") can extract and group commands of any desired type for any desired antenna and print in one of several ways.

(b) telposN.log: This is generated routinely by all observing programs. There will be as many files as there are subarrays - telpos1.log, telpos2.log, etc. Typically, there will be one record every second (or perhaps every 2 seconds). Its structure will be as follows:

HEADER RECORD
DATA RECORDs
TRAILER RECORD
HEADER RECORD for second session
DATA RECORDs for second session
TRAILER RECORD for second session
etc., for subsequent sessions.

HEADER RECORD is signified by the string "HDRTEL" in the first 6 bytes; TRAILER RECORD is signified by the string "EOFTEL" in the first 6 bytes. All other records will be considered as DATA RECORDs.

The contents of these records will be as follows:

HEADER: HDRTEL;date/time.start;interval;subarray_id;num_tel;tel_names
char*6 yymmddhhmmss i2 char*8 i2 #a4
(semicolons are indicated above as field-separators for clarity; there will not be any field separators in the actual record. Interval is the time interval between successive records in seconds)

DATA: timeseq;desired azim;desired elev;(flag;az err;el err) for each antenna
(long int) int int int int int
where timeseq is the number of seconds from start, azim,elev are in degrees, az err and elev err (true-desired) are in encoder units.

TRAILER: EOFTEL;date/time.end;interval;subarray_id;num_tel;tel_names

If all 30 antennas are active in a subarray, and if interval is chosen to be 1 sec, there will be 150 bytes each in HEADER and TRAILER records while the DATA records updated every interval (1 sec in this example) will have 186

bytes each. Possible values of interval are 1 sec, 2 sec or 5 sec.

This file will be interpreted and printed in various ways by the program "rd_tel".

(c) mon#.log: This file will contain all the monitored data base for a given subarray - mon1.log corresponding to sub_array_1, mon2.log to sub_array_2 etc. Like telpos#.log, this also has HEADER and TRAILER records and a number of data records in between. The structure of these records is as follows:

HEADER: HDRMON;date/time.start;interval;sub_array_id;num_tel;
list_of_tel;num_mon;list_of_mon_addresses

DATA: timeseq;Byte-values of each monitor_address, for each telescope.

TRAILER: EOFMON;date/time.end;interval;sub_array_id;num_tel;
list_of_tel;num_mon;list_of_mon_addresses

where timeseq is again the number of seconds since date/time.start.

This file is read and printed in various formats by the program "rd_mon."