Online filling of VLA data using the AIPS task FILLM

Gustaaf van Moorsel & Wes Young
14 February 1995

1 – Introduction

Online filling of VLA data is now available both at the VLA site (workstation miranda) and at the AOC both on the Sparc Stations and on the IBMs. Online (or realtime) FILLM is very similar to the more familiar offline FILLM. The main difference is that offline FILLM reads from a tape drive, whereas online FILLM receives data directly from the VLA Modcomp computer, though it is led to believe it is getting its data from a tapedrive, too. Depending on the workstation, one or more tape drive numbers are reserved for online use; at AIPS startup a list of all available tape drives - including the online "pseudodrives" are listed. Set **INTAPE** to a number listed as being reserved for online data filling, and a subsequent **MOUNT** connects the user to the Modcomp. If there is no realtime FILLM tape device listed at startup, ask one of the AIPS support people to help you. Before starting FILLM, make sure you run the correct version by typing:

```
version 'online'
```

2 – Why fill data online?

There are two distinct advantages of filling your data online:

1 - In the offline case, one first has to wait for the tape to arrive from the VLA site, and then for FILLM to fill the data to disk. In the online case, the data are available for inspection and reduction the second the observations finish. If something should go wrong while filling online, or if one wants to fill using different parameter values, there is always the offline route as backup.

2 - In the offline case, the files created can not really be used before FILLM finishes. At any one time, FILLM may choose to append to files created during the run, and this may fail if they are being used for something else. In the online case, the user has the option to close the files currently being written to, and create new files instead. This frees the old files for use with AIPS. The user has full interactive control over whether data is appended to existing files or new files are created.

3 – Interactive control

The user can determine interactively whether new data should be appended to existing files, or be put in newly created files. This is regulated by the first element of the hidden adverb **APARM**. 'Hidden' means that **APARM** can only be specified using the **TELL** mechanism during the execution of the task: it will not show up in **INPUTS FILLM**, but it will in **SHOW FILLM**. Both on- and offline FILLM start with the option to append data to newly created files (or older files, if **DOCONCAT=1**), whenever possible. This behavior can now be changed interactively using **TELL FILLM** according to the following table:

APARM(1)	OPTELL	procedure	meaning
	QUIT	quit	finishes current scan, then quits
0	CHAN	few	appends scans to existing data whenever possible
1	CHAN	many	creates a new file for each scan
2	CHAN	break	closes files after current scan, then continues in **few** mode
3	CHAN	stop	like quit, but does not wait for current scan to end

In the previous table, the listings under 'procedure' are short AIPS procedures which execute the TELL command and give it the proper OPTELL and APARM(1). Perhaps the most useful option is break. This closes the file when the current scan is finished, creates a new file, but after this appends further new data to this new file. The closed file is ready to be viewed and analyzed by AIPS software. This might well become the standard way of running online FILLM: let FILLM append until the user tells FILLM to close the file and create a new one. This new file will receive all the new data until the users "tells" FILLM again to break.

4 – Ending FILLM

There are three ways to end FILLM. abort is instantaneous, and will probably leave the output files in a corrupted state. The other two possibilities properly close the FILLM output: stop takes as long as the TELL command takes to be recognized, at the most 1 - 2 minutes. quit allows FILLM to finish the current scan first, which may take much longer. Finally, FILLM will end automatically under certain conditions, described in the following section.

5 – VLAOBS and subarrays

FILLM may be started ahead of time, before the proposal code of interest has begun. FILLM's starting and finishing is governed by the adverbs VLAOBS — specifying the program code for the observation to be filled — and CPARM(6) (subarray number) as listed in the following table:

VLAOBS	CPARM(6)	action
non-blank	0	FILLM starts when VLAOBS begins in any subarray. That subarray becomes the current subarray FILLM ends when a different program code in that subarray starts
non-blank	> 0	FILLM starts when VLAOBS begins in the specified subarray FILLM ends when a different program code in that subarray starts
blank	0	FILLM loads data from all program codes in subarray 1 FILLM has to be stopped interactively.
blank	> 0	FILLM loads data from all program codes in the specified subarray FILLM has to be stopped interactively.

The first case, VLAOBS non-blank, CPARM(6) = 0, is the recommended mode of running online FILLM: the user does not have to know about the actual subarray, and FILLM exits automatically. There is one caveat: under some circumstances, after program code VLAOBS has executed, there may not be another program starting in the same subarray for a long time. In that case FILLM does not finish automatically, and the user will have to stop FILLM by hand. Make sure to enter the correct program code: if you specified proposal code AV65 in the observe file, FILLM will look in vain for e.g. VLAOBS='AVO65'.

6 – How it works

A process runs on the Modcomps at the VLA site that sends visibility data to miranda (a SparcStation II at the VLA site). This visibility data is put into a shared memory buffer. On-line FILLM on miranda (or workstations at the AOC) access the shared memory via pseudo-tape routines that "fool" FILLM into thinking it is getting data from tape. On-line FILLMs in the AOC receive visibility data from a data server in the AOC (currently something.aoc.nrao.edu).

A summary of what goes into getting "real-time" data into FILLM:

- your workstation
    1. FILLM spawns an `aoc_clnt`
    2. `aoc_clnt` requests data from visibility server

- visibility server
    1. if no `aoc_srvr` is running it spawns both `aoc_srvr` and `vla_clnt`
    2. `vla_clnt` requests data from miranda

- miranda
    1. when requests for data comes from visibility server at the aoc it spawns a `vla_srvr` which sends data down to the visibility server.
    2. data transfer between the Modcomps and miranda is continuous

When the link is fully in place, data travels

1. from Modcomp
2. to miranda
3. to something
4. to your workstation.

This setup has been reasonably robust, especially against crashes or reboots of the Modcomps.

7 – Example


```
>INTAPE 2                                             listed as realtime drive at AIPS startup
>mount
AIPS 1:  Mounted on-line tape device on local host
>version 'ONLINE'
>vlaobs ' '                                           fill current observing program
>go fillm
AIPS 1:  Found in Version=ONLINE
FILLM: Task FILLM (release of 15JUL95) begins
FILLM: UV data will be written in compressed format
FILLM: Welcome to real-time FILLM!
FILLM: Shadow flag limit = 2.500E+01 meters.
AIPS 1:  Resumes
FILLM: MCINI: Processing Correlator Code ' ' with 27 antennas.
FILLM: MCINI: Processing Correlator Code ' ' with 27 antennas.
FILLM: Program = AK383 ; Tape revision number = 24.
FILLM: Create 14/02/95 .L BAND. 1 (UV) on disk 2 cno 1
FILLM: Ref.  date = 14/02/95 A-C = 1.464900 B-D = 1.385100 GHz
FILLM: FLMFQ: FQ entry tolerance = 1.000D+02 1.000D+02
FILLM: Found 1857+039 :  1 50.000 MHz at IAT 0/ 16 45 0.
>stop                                                 issues TELL command for graceful exit
AIPS 1:  Found in Version=ONLINE                      AIPS' reaction to TELL command
FILLM: has been told to change parameters             FILLM's reaction to TELL command
FILLM: Read 1404 visibilities from 1 files
FILLM: Appears to have ended successfully
```

8 – Common problems and their causes

Problem: On starting FILLM, you receive the following message:

```
FILLM1:   ZTPOPN: NO SUCH LOGICAL DEVICE = AMTO2:
FILLM1:   TAPIO: ERROR 3 OPENING FILE
```

Solution: you have not mounted a tape drive. Find out a correct tape number and do a **MOUNT**.

Problem: On starting FILLM, you receive the following message:

```
FILLM1:   ZTPWAT: ERROR 3 FOR LUN = 32 IBUFF = 1 TO TAPE
FILLM1:   ZERROR: IN ZTPWA2 ERRNO = 5 (I/O error)
FILLM1:   This usually means PARITY ERROR
```

Solution: You are running regular FILLM. Set the AIPS version to **ONLINE** and restart FILLM.

Problem: While running FILLM, commands like **stop** and **break** are not recognized.

Solution: Do a **RESTORE 0** and try again.