ON-THE-FLY SCANS WITH THE NEW SINGLE DISH CONTROL SOFTWARE

S. Righini

INAF-IRA, Istituto Nazionale di Astrofisica
Istituto di Radioastronomia, Bologna, Italy

IRA 425/08

Introduction and general overview

A new single-dish control software is being produced to serve both the Sardinia Radio Telescope and the Medicina 32m parabola. The SRT version is called Nuraghe, while the Medicina "flavour" is named ESCS (Enhanced Single-dish Control System). The overall system requirements of ESCS are described in the IRA Technical Report n. 409/07, while details on the software design and structure are given respectively in reports n. 423/08 and n. 424/08.

This report describes in detail the Nuraghe/ESCS component created to perform on-the-fly scans (OTF), and shows the results of simulations and in-antenna tests.
Prior to this development, the Medicina dish control system (the VLBI Field System) was not provided with tools to perform scans. Single observers who needed to use the antenna in OTF mode had to develop custom solutions, not integrated in the general system.

OTF is an ACS (ALMA Common Software) component aimed at computing the sequence of pointings which constitute a single line of a scan. For example: given two positions in the sky, specified by means of equatorial (J2000.0), horizontal or galactic coordinates, plus parameters such as the scanning duration, geometry and start UT time, the OTF component produces the list of UT-based coordinates to be commanded to the antenna in order to perform a smooth scan. This first release of the component allows to execute scans in the already mentioned coordinate frames, along three possible geometries: constant latitude, constant longitude or great circle. There is a further option, mainly devoted to pointing calibration scans: in this case the antenna performs horizontal scans across a sidereal position.
The route computation also rules - following user-set parameters - the execution of the pre-scan acceleration ramp and the post-scan deceleration ramp, so that the whole antenna path is deterministically known in advance.

Off-line simulations have shown that the scan computation is performed correctly, and in-antenna tests demonstrated that very fast scans (up to 20°/min) can be executed keeping a proper pointing accuracy (0.1 HPBW) even with the narrowest beam available on the Medicina dish (1.8 arcmin @ 22 GHz).

On-the-fly scans: concept

On-the-fly scanning is an observational method which dramatically reduces "dead times" and increases the antenna throughput, allowing to fast span a large area of the sky. Data acquisition, in fact, takes place while the antenna is moving along a precisely computed path across the sky. The faster the scan, the greater the stability of the system and of the environmental conditions (such as weather parameters) during the acquisition.

The maximum speed at which the scan can be executed depends, of course, on the mechanical limits of the instrument and on the antenna sensitivity: the observer must compute how much integration per beam is needed to reach the desired flux limit, and consequently determine the proper scan speed and the data sampling rate – which, in turn, must meet the data acquisition system requirements.

In the ESCS system, we define "subscan" the elementary acquisition, a sequence of which constitutes the "scan", i.e. the total observation. In a classic OTF mapping, a subscan corresponds to a single "line" of the map. During the observation design and planning, the subscan inter-spacing must be fixed considering that the final map must at least fulfil the Nyquist criterion.

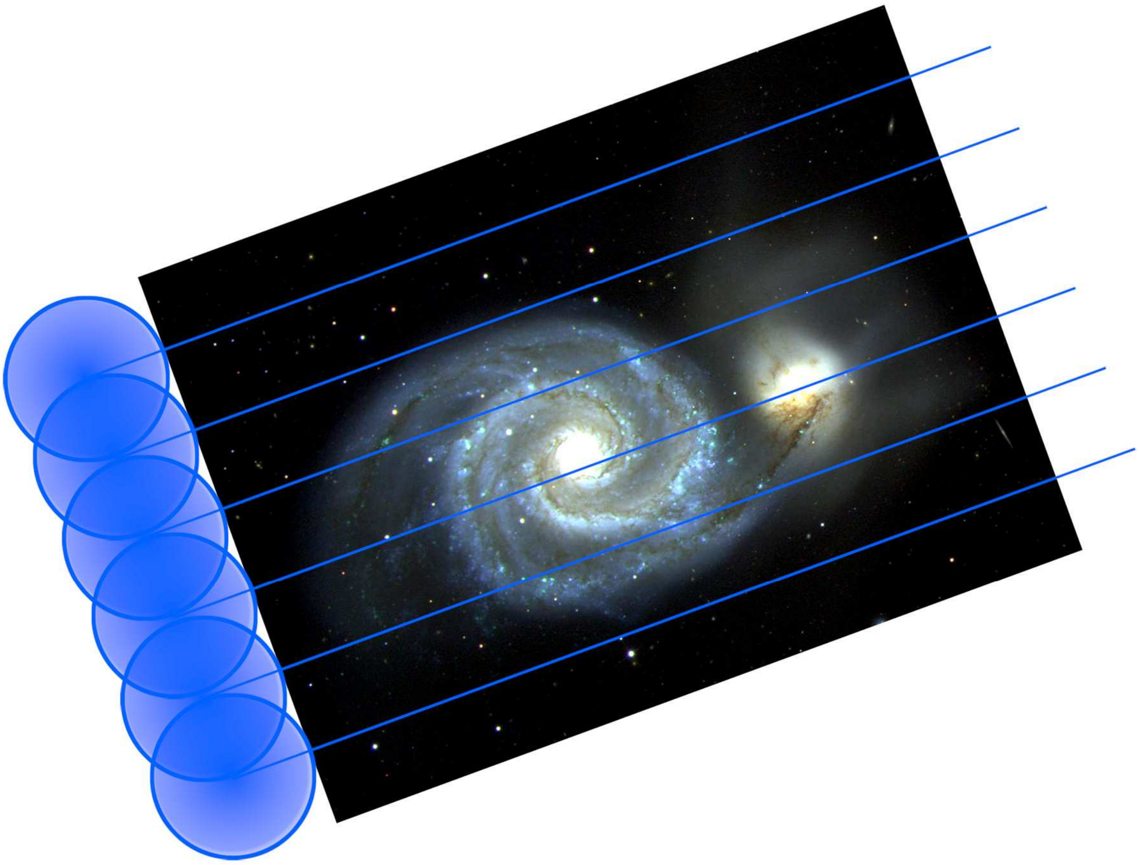


Fig. 1 Representation of an on-the-fly scan over the M51 galaxies. Each blue line shows the path of the beam centre across the sky. Adjacent lines (i.e. subscans) are spaced by HPBW/2 so that the final map is correctly sampled. Subscans can be performed in the same direction - rewinding the beam each time - or following a back-forth path in order to cut-off the slewing time.

The Nuraghe/ESCS "OTF" component

In ACS-ESCS we call "component" each system element which is in charge of a specific task. The component named OTF is responsible of computing the sequence of pointings which constitute a single on-the-fly subscan.
It gets the input values from a schedule, which is directly written by the user or can be produced – for horizontal scans only, at present - by another program elaborating the general map parameters and determining the boundaries of each subscan.
In addition to the configuration constraints, OTF obtains other parameters from AntennaBoss, the component which manages the antenna pointing and calls OTF. The following table lists the input parameters needed to set a single subscan and passed as arguments of the method called *setSubScan*.

initAz	Azimuth (radians) where the antenna is located when OTF is called. This is automatically obtained from AntennaBoss.
initSector	CW or CCW relative to the initAz value. This is automatically obtained from AntennaBoss.
initEl	Elevation (radians) where the antenna is located when OTF is called. This is automatically obtained from AntennaBoss.
initTime	UT instant (100 ns since 1582-10-15 00:00:00) when the OTF is called. This is automatically obtained from AntennaBoss.
lon1	If the description is start-stop, longitude (radians) of the constant speed scan starting point. Otherwise it corresponds to the longitude (radians) of the subscan central point. Value is obtained from user input data.
lat1	If the description is start-stop, latitude (radians) of the constant speed scan starting point. Otherwise it corresponds to the latitude (radians) of the subscan central point. Value is obtained from user input data.
lon2	If the description is start-stop, longitude (radians) of the constant speed scan stopping point. Otherwise it corresponds to the full span in longitude (radians) of the subscan. Value is obtained from user input data.
lat2	If the description is start-stop, latitude (radians) of the constant speed scan stopping point. Otherwise it corresponds to the full span in latitude (radians) of the subscan. Value is obtained from user input data.
coordFrame	Reference frame for the input coordinates. It can be equatorial J2000.0, galactic or horizontal. Value is user-defined.
geometry	Subscan geometry, it can be constant longitude, constant latitude or great circle. Value is user-defined.
subScanFrame	Reference frame for the execution of the subscan. It can an be equatorial J2000.0, galactic or horizontal. Value is user-defined.
description	Subscan definition. It can be start/stop or centre/span. Value is user-defined. Centre/span description does not apply to great circle scans.
direction	Increase or decrease, relative to the varying coordinate. Does not apply to great circle subscans.
startUT	UT instant (100 ns since 1582-10-15 00:00:00) at which the data acquisition is commanded to start. It coincides with the instant when the constant speed scanning starts. Value is user-defined.
subScanDuration	Duration of the constant speed scanning (s). Value is obtained from user input data.

The component also reads the following values, specific for the telescope in use, from the Configuration DataBase (CDB). Some of these parameters were determined for the Medicina dish by means of on-field tests:

- Maximum azimuth slewing rate allowed by mechanical constraints (°/s);
- Maximum elevation slewing rate allowed by mechanical constraints (°/s);
- Maximum azimuth scan rate allowing correct pointing (°/s);
- Maximum elevation scan rate allowing correct pointing (°/s);
- Maximum azimuth acceleration (°/s^2);
- Maximum elevation acceleration (°/s^2);
- Acceleration "Scale Factor" – scaling the az-el acceleration in order to perform smooth acceleration/deceleration ramps before/after the constant speed scan;
- Site information (longitude, latitude, elevation, etc…);
- Observation information (HPBW, DUT1, etc...).

Once the initialisation and setup are complete, OTF controls if the user-defined parameters are correct. In particular it checks that:

- values match with the available options (e.g. the coordFrame value cannot be "ecliptic" because this coordinate frame is not implemented) and that they are correct;
- values are consistent with each other. For example, a great circle geometry is not compatible with the centre/span description;
- the azimuth and elevation rates stay below the allowed limits during the whole subscan. This is particularly critical for great circle scans, since they can imply very high values for the rates in restricted sections of their curved paths along the sky – for this reason, the whole path is simulated in advance. In case the az-el speeds exceed the instrumental limits the program raises an error and ends. If, instead, the speed is below the maximum values but exceeds the limits which have been tested to be reliable for a correct pointing, then a warning is raised.

The next step is the computation of the acceleration/deceleration ramps to be appended to the constant speed scan. The user-defined start coordinates and UT time are, in fact, relative to the beginning of the constant speed path. OTF automatically computes the ramps in order to match those constraints, using the acceleration and scale factor values written in the CDB to compute a uniformly accelerated route. The coordinates and UT time relative to the first point of the acceleration ramp are commanded to start the antenna motion.

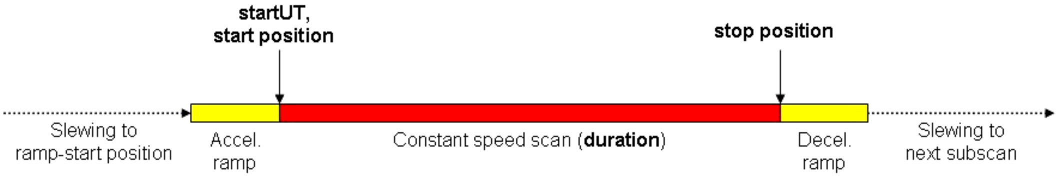


Fig. 2 Schematisation of the timeline/path of a subscan. The antenna system defines, elaborating the user-input information ore reading the schedule, the subscan boundaries and duration (bold parameters), referring to the constant speed scan (red line), which are passed to the OTF component. This adds the acceleration/deceleration ramps (yellow lines), computing their start and stop positions and the relative timings. The initial ramp start position is commanded to the antenna, which slews to those coordinates, waits for the ramp start UT time, runs the accelerated path and begins the "constant speed path" at the correct time and position.

OTF then checks if this ramp-start position can be reached in time: when called, it is passed the current UT instant and antenna position and computes if the necessary antenna slewing time is short enough to reach the ramp start coordinates within the commanded UT time. If the target is too far, a warning is raised and the general system (ESCS) cross-checks the impossibility to reach the commanded position; if the problem is confirmed, the subscan aborts and the system skips to the next one in the schedule (if any).
At this point, everything is set and the OTF component is asked to compute the sequence of pointings composing the subscan, and also to check if the actual motion correctly follows the predicted positions.

This happens by means of two methods, which are recursively called by AntennaBoss:

- *getHorizontalCoordinates* – given a certain UT instant, it gets the corresponding azimuth and elevation positions the antenna is going to be commanded to, as a function of the subscan setup. This method is called in advance, i.e. passing a UT instant which is some seconds ahead;
- *checkTracking* – given a UT instant, it checks that the corresponding true position of the antenna coincides with the previously predicted one, within 0.1*HPBW. This method is called in realtime, i.e. passing the current UT instant.

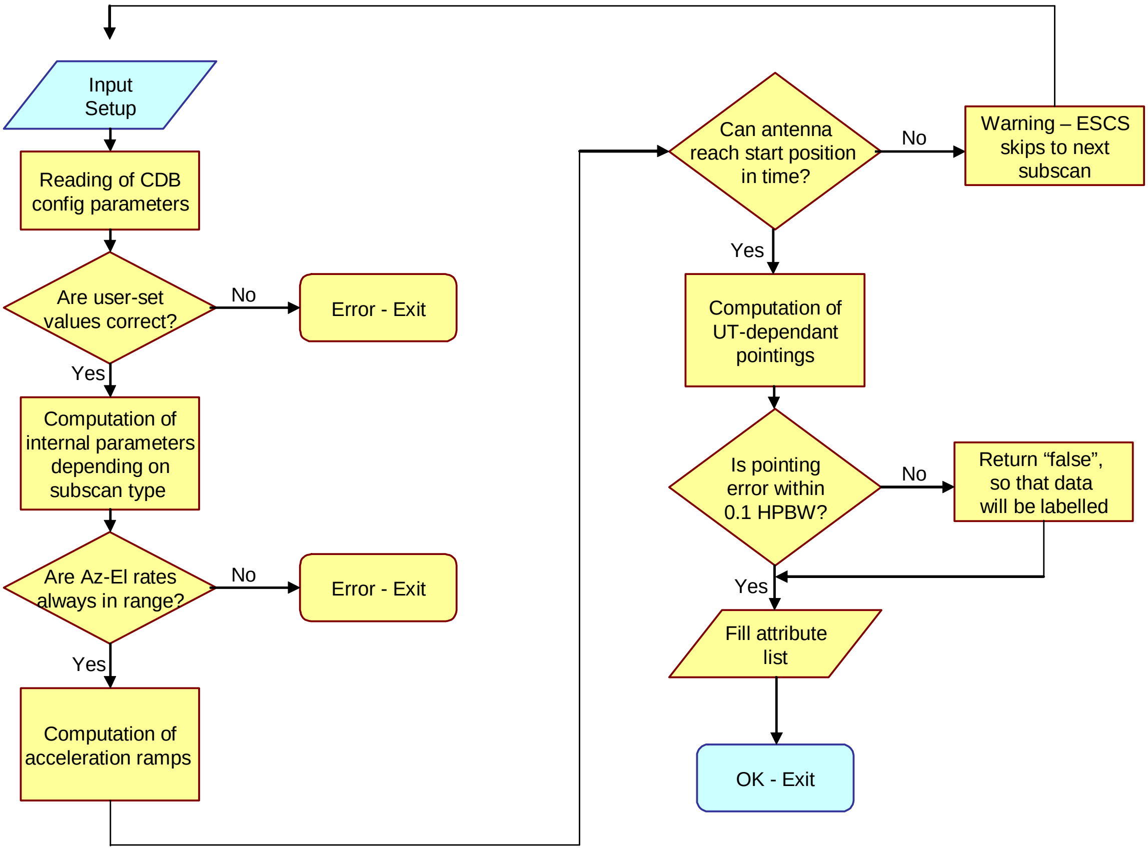
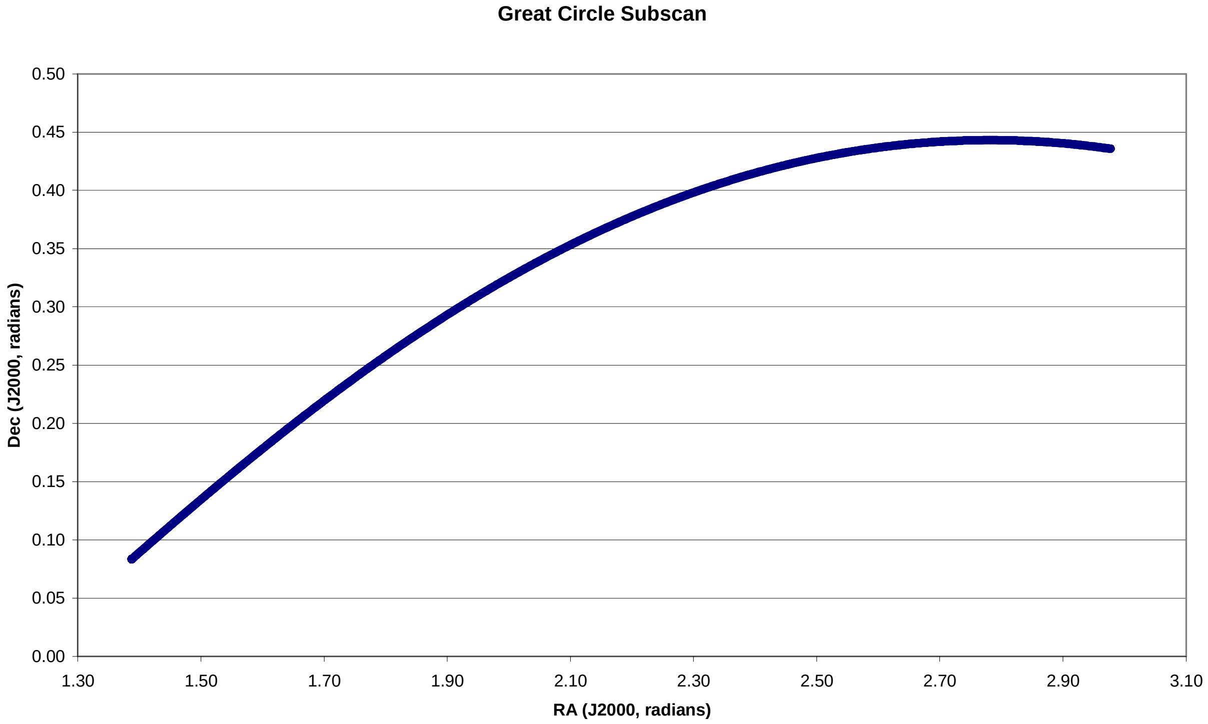

Input
Setup
Reading of CDB config parameters
Are user-set values correct?
No
Error - Exit
Yes
Computation of internal parameters depending on subscan type
Are Az-El rates always in range?
No
Error - Exit
Yes
Computation of acceleration ramps
Can antenna reach start position in time?
No
Warning – ESCS skips to next subscan
Yes
Computation of UT-dependant pointings
Is pointing error within 0.1 HPBW?
No
Return “false”, so that data will be labelled
Yes
Fill attribute list
OK - Exit

Fig. 3 Flowchart illustrating the main features of the OTF component.



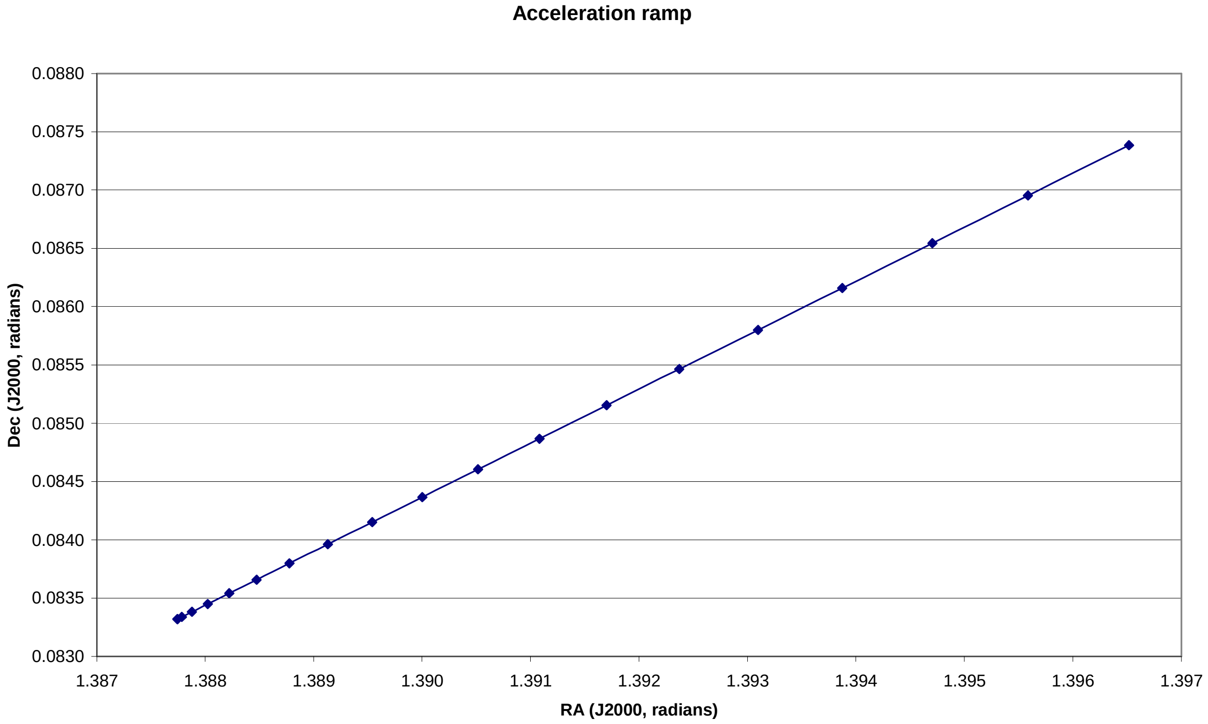
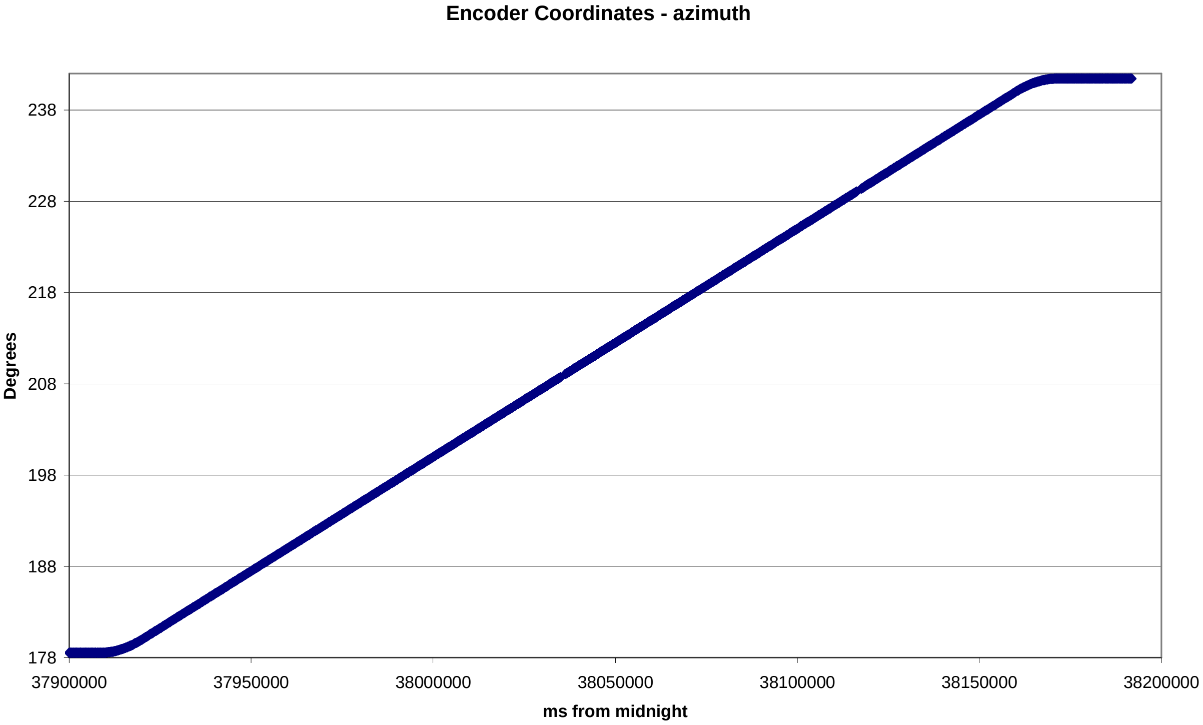


Fig. 4 Example of a great circle subscan computed by the OTF component. The top figure shows the overall sky path. The bottom plot zooms on the initial acceleration ramp path: samples are equally spaced in time, and the uniformly accelerated motion is evident. Setup parameters: startLon 80.0°, startLat 5.0°, stopLon 170.0°, stopLat 25.0°, coordFrame=subScanFrame Equatorial, description StartStop, duration 600 s.

OTF tests with the Medicina dish

Several scans were performed to test the OTF component, varying the azimuth/elevation speed and acceleration scale factor to estimate the maximum values generating acceptable pointing errors.



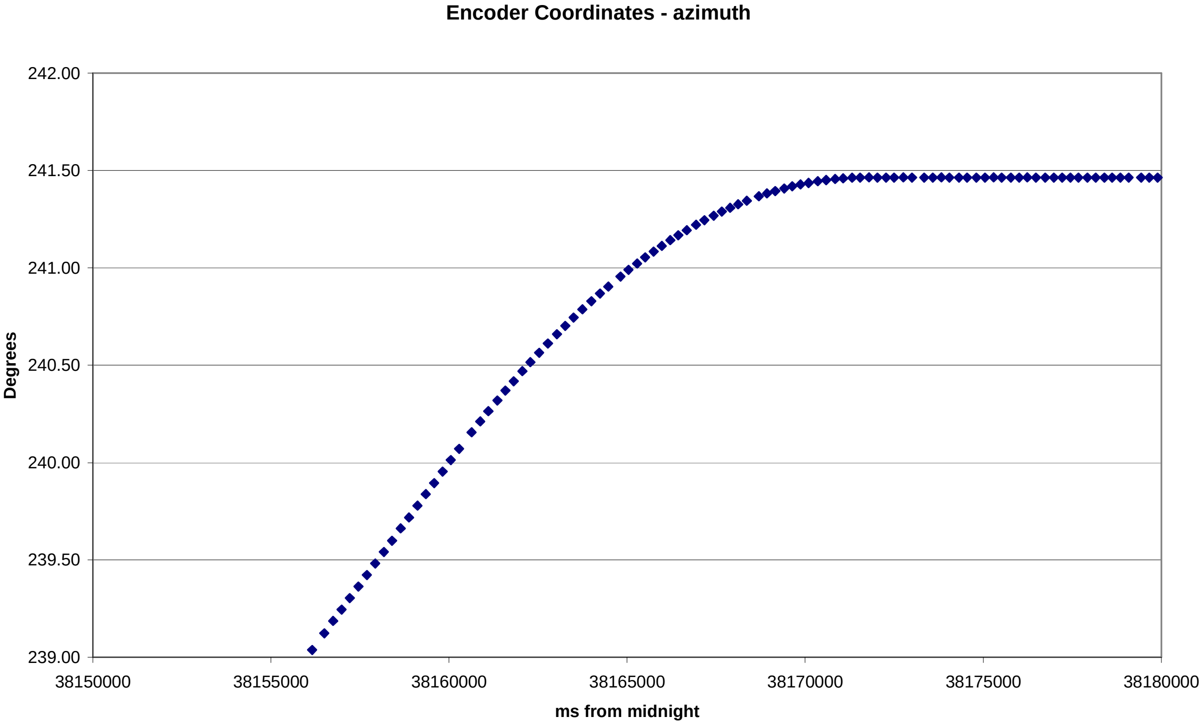


Fig. 5 Example of a real constant elevation subscan performed using the OTF component. The top figure shows azimuth vs. time for the whole path. The bottom plot zooms on the final deceleration ramp: samples are taken every 200 ms. Setup parameters: startLon 180.0°, startLat 45.0°, stopLon 240.0°, stopLat 45.0°, coordFrame=subScanFrame Horizontal, description StartStop, duration 240 s.

Since the ESCS system was not yet provided with a tool to get the commanded coordinates and the encoder ones contemporarily (i.e. relative to the same UT instant), the chosen solution was to download the encoder coordinates and the associated pointing error, which was calculated internally to the ACU (Antenna Control Unit) as the difference between the commanded and the encoder values.

This turned out to produce false values in the pointing error measure: as figure 6 shows, there were outliers appearing with a sort of regular "arrangement", highlighted by the red lines. These values (as high as 0.03 degrees = 1.0*HPBW) cannot have any physical meaning, since they would imply that the antenna was going one beamsize off-pointing and returning to the correct position in less than 200 milliseconds (the data sampling interval), which is clearly not possible.

The origin of the outlier values probably was a time delay happening inside the ACU: the commanded/encoder coordinate samples were not always relative to exactly the same UT time. Because of the fast antenna speed, their difference yielded extremely high values. In fact, travelling at 10 °/min the antenna was spanning 0.03° in a sampling interval; this way, confronting coordinates relative to different instants meant measuring an excess error up to about one HPBW.

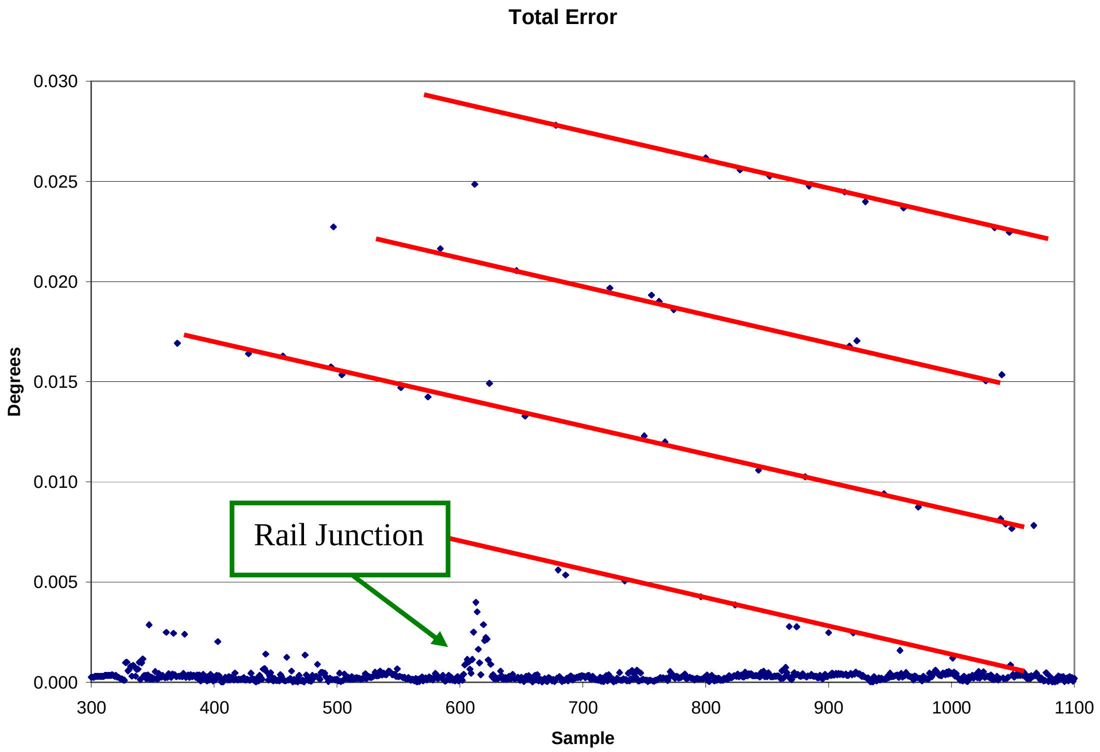


Fig. 6 Graph shows the total pointing error (difference between commanded-encoder coordinates, computed by the ACU) as a function of time. This is a single subscan performed at constant elevation (45°), moving the antenna along azimuth only, from 180° to 220°, with a data sampling interval of 200 ms. The "bump" appearing after sample #600 is due to the antenna mount wheels passing on the rail junctions. Red lines highlight the recurrent presence of outlier values.

Data were cleaned from the outlier values using a 3-point median filter. This led to the almost complete removal of the inconsistent measures, while preserving true features like the "bump" appearing around sample #250 (figure 7), produced when the antenna mount wheels revolve on the base rail junctions.

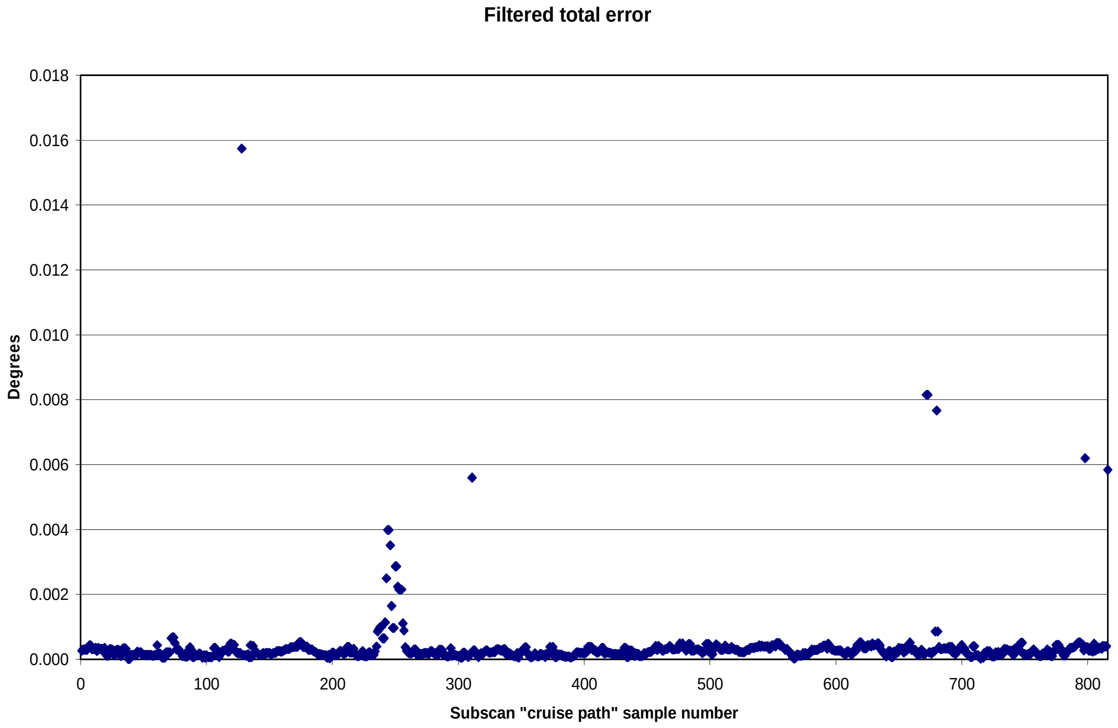


Fig. 7 Values after the application of a 3-point median filter. Data were cropped: the graph shows the constant speed path only. Notice the change in the y-axis scale; the few outlier values which were not removed are lower than 0.5*HPBW.

The following table illustrates some examples of the final statistics on the measured pointing error, given as a fraction of the HPBW. Many of these tests focused on azimuth scans, since the first application of the new system will be to carry out a survey using this peculiar observing method.

DOY	Coord. Frame	Start Lon-Lat	Stop Lon-Lat	Geom	Durat	Scale Factor	Cruise Speed	Cruise Speed	Mean	3σ
		°	°		s		°/m Az	°/m sky	%HPBW	%HPBW
199	**EQ**	**170 - 0**	**190 - 40**	**GC**	**480**	**30**	**-**	**~ 5.5**	**1.64**	**5.11**
217	HOR	180-45	220-45	LAT	240	30	10	7.0	1.06	8.86
217	**HOR**	**180-45**	**220-45**	**LAT**	**192**	**30**	**12.5**	**~ 8.8**	**1.25**	**11.25**
217	HOR	180-45	220-45	LAT	100	30	24	~ 16.8	1.95	23.65
217	**HOR**	**180-45**	**220-45**	**LAT**	**240**	**50**	**10**	**7.0**	**1.07**	**10.03**
217	HOR	180-45	220-45	LAT	192	50	12.5	~ 8.8	1.01	7.41
217	**HOR**	**180-45**	**220-45**	**LAT**	**100**	**50**	**24**	**~ 16.8**	**1.73**	**11.34**
217	HOR	180-45	220-45	LAT	240	10	10	7.0	1.08	8.86
217	**HOR**	**180-45**	**220-45**	**LAT**	**192**	**10**	**12.5**	**~ 8.8**	**1.21**	**8.55**
217	HOR	180-45	220-45	LAT	100	10	24	~ 16.8	2.10	25.01
248	**HOR**	**180-45**	**240-45**	**LAT**	**240**	**10**	**15**	**~ 10.4**	**1.90**	**11.79**
248	HOR	180-45	240-45	LAT	240	15	15	~ 10.4	2.60	10.38
248	**HOR**	**180-45**	**240-45**	**LAT**	**240**	**30**	**15**	**~ 10.4**	**1.94**	**5.88**

The above table shows that an azimuth scan speed of 24 °/min produces a stable antenna pointing only performing very slow and long acceleration ramps (i.e. for high "scale factor" values, see tables below).

Ideal values to combine quite short ramps with fast scanning are:

- scale factor = 15 (yielding ramps less than 10 seconds long)
- constant cruise speed = 15 °/min (@ Elevation=0°)

Nominal duration of each acceleration/deceleration ramp
for the different scale factors and cruise speeds
(azimuth scans)

Scale Factor	**Azimuth Acceleration**	**Resulting Az Ramp Duration (s)**			
	°/s^2	*Cruise speed (°/min)*			
		5	***10***	***15***	***24***
10	0.040	2.08	4.17	6.25	10.00
15	**0.027**	**3.13**	**6.25**	**9.38**	**15.00**
20	0.020	4.17	8.33	12.50	20.00
30	**0.013**	**6.25**	**12.50**	**18.75**	**30.00**
50	0.008	10.42	20.83	31.25	50.00
100	**0.004**	**20.83**	**41.67**	**62.50**	**100.00**

Sky extension of each acceleration/deceleration ramp
for the different scale factors and cruise speeds
(azimuth scans)

Scale Factor	**Azimuth Acceleration**	**Resulting Az Ramp Extension (°)**			
	°/s^2	*Cruise speed (°/min)*			
		5	***10***	***15***	***24***
10	0.040	0.09	0.35	0.78	2.00
15	**0.027**	**0.13**	**0.52**	**1.17**	**3.00**
20	0.020	0.17	0.69	1.56	4.00
30	**0.013**	**0.26**	**1.04**	**2.34**	**6.00**
50	0.008	0.43	1.74	3.91	10.00
100	**0.004**	**0.87**	**3.47**	**7.81**	**20.00**

The schedule reader

ESCS was also provided with a "scheduler". This component reads a user-set schedule, which consists of 4 files, in order to perform a sequence of LST/UT-based OTF subscans – or commands for other tracking modes, which are not covered by this document.

- ***FileName*****.bck** → This file lists the configuration parameters for the backend. The content is organised as in the following examples:

```
STD:BACKENDS/MFTotalPower {
        integration = 40
        calSwitch = 0
        configure = *,-1,330.0,0.000025,10.0,-1,-1
        enable = 1,1,1,1,1,1,1,1,1,1,1,1,1,1
 }

HIGH:BACKENDS/MFTotalPower {
        integration = 25
        calSwitch = 2
        configure1 = 1,-1,2000.0,0.000025,9.0,-1,-1
        configure2 = 2,-1,2000.0,0.000025,9.0,-1,-1
        enable = 1,1,0,0,0,0,0,0,0,0,0,0,0,0
 }
```

The first field is the procedure name and must be unique in the file, then follows the name of the backend instance the procedure refers to. The open bracket must lie on the same line. Between brackets the configuration commands for the backend are given. They must comply with what the backend expects, so they may vary from case to case. The above examples refer to the new continuum backend, but details are not included in this document. The scheduler will not start any acquisition until the whole configuration procedure is finished.

- ***FileName*.cfg** → This file lists the configuration parameters for the frontend (subreflector position, local oscillator frequency, receiver setup, etc…). In the present implementation it is not used and is left empty;

- ***FileName*.lis** → This is the list of the subscan configurations. It specifies the setup parameters for a series of subscans labelled with an ID number. Here is an example of how the file must be written (with TAB-separated values):

```
#ID   type   lon1     lat1     lon2     lat2    frame   sFrame   geom   descr   dir    duration
  1   OTF    15.500   23.000   18.000   27.000  EQ      EQ       GC     SS      INC    240.000
  2   OTF     0.000    0.000   30.000    0.000  GAL     GAL      LAT    SS      INC    120.000
  3   OTF    45.000   45.000   20.000    0.000  HOR     HOR      LAT    CEN     DEC     90.000
  4   OTF    45.000   45.000   20.000    0.000  HOR     HOR      LAT    CEN     INC     90.000
```

The first line, introduced by a "#" symbol, is a comment line and is ignored by the system; it contains the column labels, which correspond to the keywords listed in the previously presented table. In particular:

lon1, lat1, lon2, lat2 = same as the identically named keywords,
but here the values are in degrees
frame = coordFrame (EQ, HOR or GAL)
sFrame = subScanFrame (EQ, HOR or GAL)
geom = geometry (LON=constant longitude, LAT=constant latitude, GC=great circle)
descr = description (SS=start-stop, CEN=center-span)
dir = direction (INC=increase, DEC=decrease)
duration = subScanDuration (seconds)

Then the subscan list follows.

Subscan 1 is a great circle scan performed in equatorial J2000.0 coordinates, starting from the position (RA_1,DEC_1)=(15.5°,23.0°) and ending at position (RA_2,DEC_2)=(18.0°,27.0°), lasting 4 minutes. The indication of direction (here: INC) is necessary but is not considered by the system, since it is meaningless for this scan geometry.

Subscan 2 is a constant latitude scan in galactic coordinates. It is specified by means of the start and stop position - in increasing direction - which respectively are: (l_1,b_1)=(0.0°,0.0°) and (l_2,b_2)=(30.0°,0.0°). Subscan is performed in 2 minutes.

Subscan 3 is a constant latitude scan in horizontal coordinates, i.e. an azimuth scan. It is described by means of a central position and a span: the central position is (Az,El)=(45.0°,45.0°), and the span is 20.0° in azimuth. Considering that the direction is specified as "decrease", this means that the scan start position will be (Az,El)=(55.0°,45.0°) while the scan stop position will be (Az,El)=(35.0°,45.0°).

Subscan 4 is identical to subscan3, but is performed in the opposite direction.

- ***ScheduleName*.scd** → This is the schedule, i.e. the operating list of time-based subscans. This file must begin with a header written as follows (keywords and values are TAB-separated):

```
PROJECT:     ProjectName
OBSERVER:    ObserverName
SCANLIST:    FileName.lis
MODE:        UT
CONFIGLIST:       FileName.cfg
BACKENDLIST:      FileName.bck
```

The "MODE" keyword refers to the user choice concerning the schedule timing. Its value can be LST (local sidereal time) or UT (universal time). If the LST timing is chosen, the user must specify after the label, with a tab-separated integer, the number of times the schedule is to be cyclically run.

After the header, there is a comment line indicating the column labels, followed by the schedule rows, filled with TAB-separated values:

```
#ID     startUT         duration      gap     scan#   config  backend
   1    12:00:00.000    240.00        10.000  1       -1      NULL
   2    12:05:00.000    240.00        10.000  1       -1      NULL
   3    12:10:00.000    240.00        10.000  1       -1      STD:MANAGEMENT/Writer1
   4    12:15:00.000      90.00       10.000  3       -1      HIGH:MANAGEMENT/Writer1
   5    12:16:40.000      90.00       10.000  4       -1      NULL
   6    12:19:30.000    120.00        10.000  2       -1      NULL
```

After the line ID, the start time is given (hh:mm:ss.sss), followed by the subscan duration (always in UT seconds) relative to the constant speed path. The "gap" value is the time interval – in UT seconds – that ESCS waits, after the completion of the constant speed scan, before commanding the antenna to the next subscan. In practice, it must be equal to the duration of the deceleration ramp plus a short "safety overhead".

If a certain antenna slewing is necessary between two consecutive subscans, it is not to be included in the "gap" value. Instead, it must be taken into account when determining the startUT/startLST instant of the following subscan.

Column “scan#” indicates the ID number of the subscan to be executed, as listed in file FileName.lis.

The config column indicates where to find the configuration parameters for the receiver: -1 stands for the present implementation, which does not allow the use of the configuration file and relies on the setup performed externally.

The last column is devoted to the backend setup, if the value is “NULL” the data recording will not take place, otherwise the string specifies the backend procedure and the target – i.e. the component actually writing the output, in the form

PROCEDURE:TARGET

In the above example, the string STD:MANAGEMENT/Writer1 means we are calling the STD procedure given in the .bck file and using Writer1 (under the Management subsystem) as the data writer.

Focusing on lines 4 and 5: these are scans sweeping the same azimuth strip in opposite directions, which means that the first scan stops exactly where the second must begin. Therefore there is no slewing between the two, and the second scan begins right 100 seconds (duration+gap) after the first one.

Please notice this is only a first version of the schedule, further developments and changes will be described in the reports to come.