Checking the Yebes 13.2 m RAEGE antenna servosystem

P. de Vicente

Informe Técnico IT-CDT 2016-27

Revision history

Version	Date	Author	Updates
1.0	23-10-2013	P. de Vicente	First draft
1.1	01-12-2016	P. de Vicente	Review and completion of sections

Contents

1 Introduction

This report describes some preliminary tests performed at the 13.2 m antenna in 2013 to verify the operation of the control system using remote commands. All tests were done without a receiver and they rely on the analysis of variables from the ACU status and on the visual inspection of the Local Control Panel. Some of the tests yielded unsuccesfull results which helped to debug the ACU (MT-Mechatronics) and the remote (Yebes) software. These errors were corrected by the time of the measurements and hence they are not reported here.

The report was originally written in 2013 but it was not complete. The author has reviewed it in 2016 and completed some sections and missing figures from the first version.

2 A simple overview on the remote operation

Remote operation is achieved by sending commands through a TCP/IP socket to the ACU at port 9000. All commands send back an acknowledgement stating if the command has been acepted or rejected. The acknowledgment may take these values:

- (1) command done (accepted and executed)
- (4) wrong mode
- (5) parameter error
- (6) wrong command length
- (7) undefined command
- (9) command accepted
- (10) command timeout

and it also returns the serial number of the command, command number since the connection was started, command identifier (150) and additional information like the header and foot and the message length (see the ICD for a complete information).

The acknowledgement takes between 300 and 900 ms to be received after the command is sent. The status information of the antenna is sent every 200 ms on port 9001 through a TCP/IP socket. The Interface Control Document (ICD), describes the commands, its format and their goal as well as the status words reporting on the current state of the antenna. The version of the ICD by the time of the tests was dated 18 October 2013. Newer versions have been delivered later, being the last one from 2015.

Since the telescope did not have a receiver, all tests were done inspecting the acknowledgement for each command and the status words received every 200 ms. No astronomical observations with data recording were available.

Our first remote connection was unsucessful and required minor recabling: the ethernet cable was connected to Beckhoff switch CU2008 and no traffic was observed in the ports where green cable WC102 cable was connected. Cable WC102 which connects to LCP port XC102 in the upper part of the right side of the LCP was replaced by our ethernet cable. Hence, currently, the Yebes cable is directly connected to port XC102 without going through the switch provided by MT-Mechatronics.

Figure 1: *Picture of the Beckhoff switch (to the left), the Yebes ethernet cable (white one crossing the image) and the LCP from behind. The connection panel can be seen to the right*

3 Time reference

The ACU needs an external timing reference to keep its time to UTC and this is provided by an IRIG-B signal. We provide this signal from a GPS receiver installed in the 40 m backends room and transmit it using an optical fiber. Since the IRIG-B is a 10 KHz signal we use a Low Frequency Optical Transmitter and Receiver from ViaLite located at each end of the optical fiber. At the backends room we use two different GPS receivers, a TrueTime XL-DC and a Symmetricom XLi located at the 40 m backends room. Both of them provide an IRIG-B signal output.

Since several units at the 40 m radio-telescope require the IRIG-B for synchronization we use a Signal Distribution Unit from Meinberg that provides several outputs. However the output from the distributor does not work properly for the ACU. Fig. 2 shows two snapshots of the clock status of the Local Control Panel: one is when connecting directly to the Symmetricom GPS receiver and the second one from the distributor.

After some investigation we found that the Meinberg distributor presents an impedance of 600 Ohms in all its outputs, whereas the signal from the GPS receiver is 50 Ohm. We have made a modification inside the distributor to change the impedance of one of its outputs so that the signal is properly adapted and does not loose power.

3.1 Checking ACU time and DUT1 implementation

The status message of the ACU distributes the UTC time every 200 ms as a Modified Julian Day and the antenna notification channel reads it and redistributes it again inside a structure

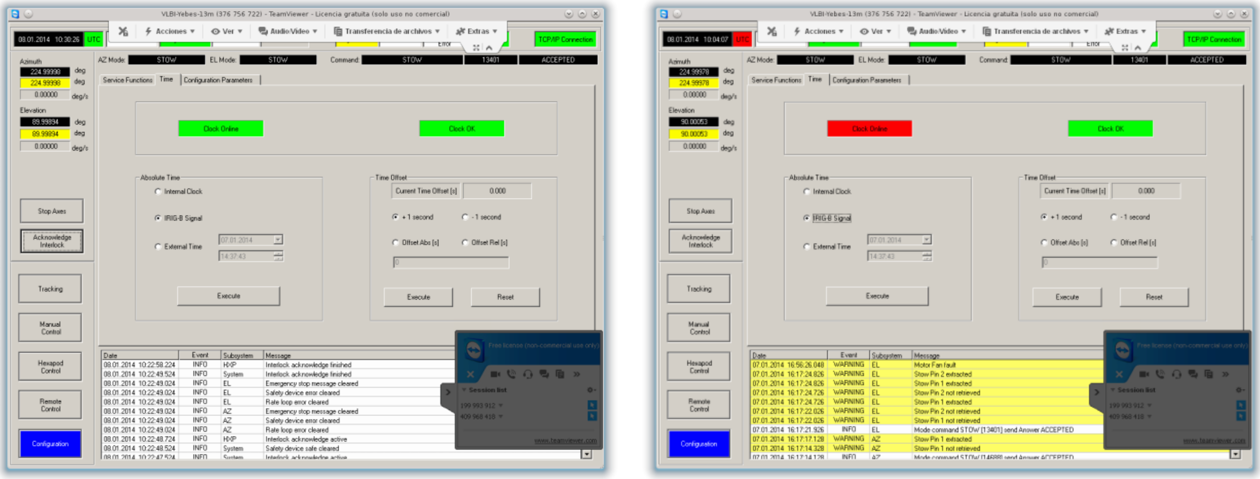

Figure 2: *Snapshot of the clock status from the Local Control Panel using the GPS receiver IRIG-B output (left) and a Timetech IRIG-B distributor.*

which also contains status information of the antenna like Az and El, errors in Az and El, the subreflector position and if the antenna is tracking. This redistributed time is called RAEGE time and it is used all over the copmutations inside the higher level control system including the data aquisition. We have made a comparison among the different time systems.

Fig. 3 shows two comparisons with a sampling period of 10 ms:

- Time distributed by the ACU minus NTP (Network Transfer protocol) at the control computer. The time at the control computer is set using NTP and its error is a few milliseconds at most. The NTP is recieved from the yebes GPS receiver which acts as a stratum 1 server.
- Time by the ACU minus time distributed by the notification channel (official RAEGE time).

According to Fig. 3 the ACU time differs from the GPS time obtained by NTP at most 4 ms. The saw-tooth pattern that we see is due to the ACU time being updated every 200 ms. Inside the ACU the time runs smoothly. The ACU time matches the GPS time exactly when it is published and hence it does not correspond to an average between the previous and the last status message. On the other hand the ACU time is exactly the same as the RAEGE time except for a small slot of time less than 50 ms, during which they differ 200 ms. This happens because the notification channel takes some time to update its value. We think this is an internal ACS issue. In any case this is not relevant since the real time position of the antenna distributed with the notification channel is totally bound to the time that is distributed at that same moment.

Apparently the DUT1 correction is not included in the distributed ACU time since at the time of Fig. 3 the DUT1 was -400 ms and this large quantity should have been seen when comparing to the NTP time. Furthermore we changed the DUT1 manually and did not see any change in the distributed time compared to NTP, only a small jump of 40 ms during 400 ms (two updates) and then back to the previous difference. However the DUT1 is taken into account for tracking tables (see section 7)

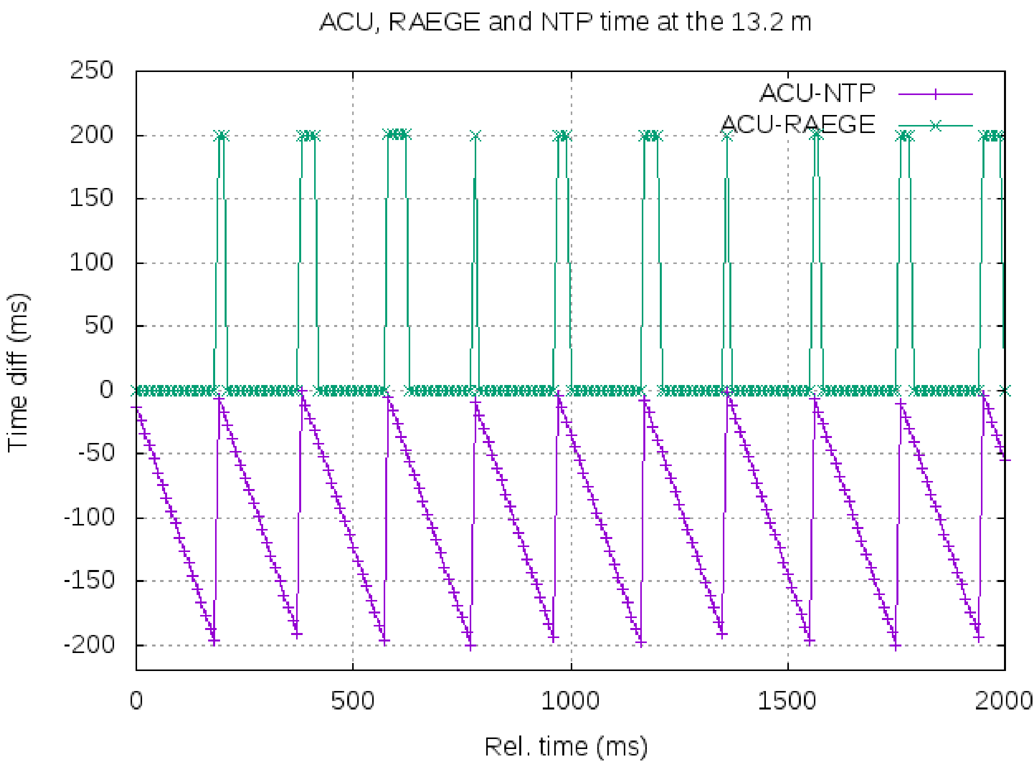

Figure 3: *Red curve: ACU time - time at the control computer synchronized by NTP. (ACU-NTP). Green curve: ACU time - RAEGE time. Time was sampled every 10 ms*

4 Basic operations: stowing, unstowing, referencing and moving the antenna

The antenna has 3 stow pins: one in azimuth and two in elevation. Apparently the software allows to operate the three independently but this seems not to be documented in the current version of the ICD. The first parameter can be 1 (for azimuth), or 3 (for both elevation stow pins), if the number is 2 only one of the stow pins in elevation will be used. Unstowing the antenna takes approximately 3 seconds. Since all stow pins are extracted at the same time, this timeout is not additive. The status of the stow pins and possible errors is obtained from a bit coded stow status word.

Below we show the timeline for the command and the status of the antenna from a log of the commands sent:

```
2013-10-25T08:43:47.825 [ACU13M - ] unstowAz() command_identifier: 2051  command_serial_
2013-10-25T08:43:47.925 [ACU13M - ] ACU ack not received in 100 ms
2013-10-25T08:43:48.025 [ACU13M - ] ACU ack not received in 200 ms
2013-10-25T08:43:48.126 [ACU13M - ] ACU ack not received in 300 ms
2013-10-25T08:43:48.226 [ACU13M - ] ACU ack not received in 400 ms
2013-10-25T08:43:48.326 [ACU13M - ] ACU ack not received in 500 ms
2013-10-25T08:43:48.426 [ACU13M - ] ACU ack not received in 600 ms
2013-10-25T08:43:48.426 [ACU13M - ] ack.command_serial_number=73, ack.status=9
2013-10-25T08:43:48.477 [ACU13M - ] unstowEl() command_identifier: 2051  command_serial_
2013-10-25T08:43:48.577 [ACU13M - ] ACU ack not received in 100 ms
2013-10-25T08:43:48.678 [ACU13M - ] ACU ack not received in 200 ms
2013-10-25T08:43:48.778 [ACU13M - ] ACU ack not received in 300 ms
2013-10-25T08:43:48.878 [ACU13M - ] ACU ack not received in 400 ms
2013-10-25T08:43:48.978 [ACU13M - ] ACU ack not received in 500 ms
2013-10-25T08:43:49.078 [ACU13M - ] ACU ack not received in 600 ms
2013-10-25T08:43:49.079 [ACU13M - ] ack.command_serial_number=74, ack.status=9
```

```
Time In1Az In1El In2El Out1Az Out1El Out2El
----------------------------------------------------------------------
08:43:51.741716  True  True  True  False  False False
08:43:51.843199  True  True  True  False  False  False
08:43:51.944806  False  True  True  False  False  False
08:43:52.046455  False  True  True  False  False  False
08:43:52.147838  False  True  True  False  False  False
08:43:52.249530  False  True  True  False  False  False
08:43:52.351348  False  True  True  False  False  False
08:43:52.452942  False  False  True  False  False  False
08:43:52.554570  False  False  False  False  False  False
08:43:52.656237  False  False  False  False  False  False
08:43:52.757772 False  False  False  False  False  False
08:43:52.859407  False  False  False  False  False  False
08:43:52.961375  False  False  False  False  False  False
08:43:53.062812  False  False  False  False  False  False
08:43:53.164478  False  False  False  False  False  False
08:43:53.266268  False  False  False  False  False  False
08:43:53.367797  False  False  False  False  False  False
08:43:53.469365  False  False  False  False  False  False
08:43:53.570799  False  False  False  False  False  False
```

```
08:43:53.672320  False  False  False  False  False  False
08:43:53.774044  False  False  False  False  False  False
08:43:53.875661  False  False  False  False  False  False
08:43:53.977170  False  False  False  False  False  False
08:43:54.078761  False  False  False  False  False  False
08:43:54.180364  False  False  False  False  False  False
08:43:54.282392  False  False  False  False  False  False
08:43:54.384146  False  False  False  False  False  False
08:43:54.485601  False  False  False  False  False  False
08:43:54.587428  False  False  False  False  False  False
08:43:54.689070  False  False  False  True  False  False
08:43:54.790745  False  False  False  True  False  False
08:43:54.892200  False  False  False  True  False  False
08:43:54.993761  False  False  False  True  False  False
08:43:55.095418  False  False  False  True  False  False
08:43:55.196964  False  False  False  True  True  True
}
```

According to the previous log, the unstow commands take 4 seconds to take effect. See how the Azimuth stow pin is commanded first and 600 ms later the Elevation stow pins are commanded out. The acknowledgement from each command is received 600 ms after the command is sent. Once the command is received by the motors, azimuth and elevation stow pins take 2.8 ± 0.1 seconds to be totally out.

At Yebes stow position is at azimuth 225° and elevation 90°. The stow operation can be performed from any position of the antenna with a single command, **driveToStow**, which drives the antenna to stow position at 1/2 the maximum speed. If a larger velocity is commanded the ACU will acknowledge with a "param error". Command, **stow**, stows the antenna, only if it is in the target position. At the 13.2 m antenna the high level routines only use the first command. Unstowing can be performed from any state of the antenna (even with the motors deactivated)

If the ACU has been reset previous to an unstow operation, the azimuth and elevation encoders need to be calibrated, or *referenced*, as MT-Mechatronics names this operation. Referencing can only be done after unstowing the antenna since the procedure requires moving the antenna. This operation takes 20 seconds for the azimuth axis and 30 seconds for the elevation one. The reference status can be obtained from the general status for each axis in the Axis Status Word for each axis. Bit 18 provides this information. If it is 1, the axis has already been referenced, if it is set to 0, no reference has been performed and a calibration is required previous to any movement. When the axes have not been referenced the Local Control Panel (LCP) displays number 500 both in the azimuth and elevation frame with the background in yellow.

Referencing may fail if the elevation encoder, a tape laid on one of the sides of the azimuth cabin (see Fig. 4) is dirty with oil or if water vapour has condensed. In this case one or more reading heads may have trouble reading the marks on the tape. This has happened at Yebes several times. To avoid water condensation a permanent air conditioning system was installed inside the azimuth cabin. Oil spills have been avoided with some metalic straps that prevent the oil dropping onto the tape.

All tests checking stow and unstow operations completed successfully. The reference operation also worked correctly, except in the cases mentioned above and which have already been corrected. Referencing the axes can be done at any elevation or azimtuh position and we have

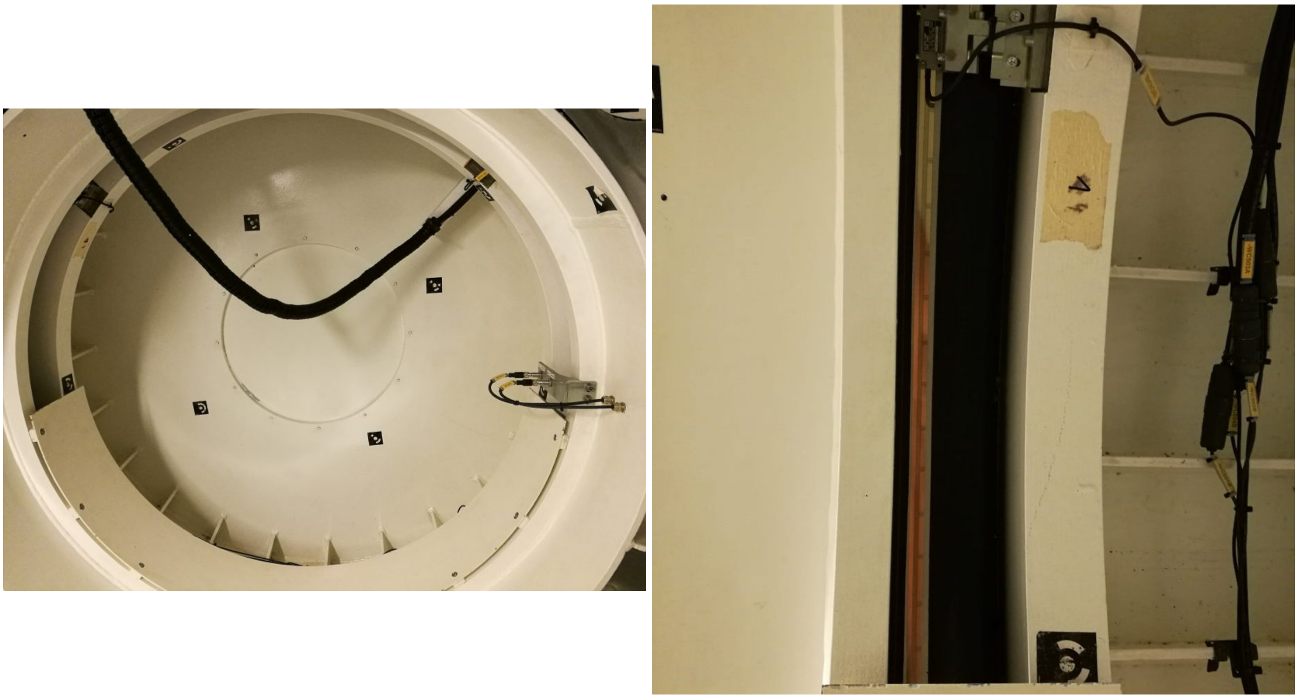

Figure 4: *Elevation encoder tape and reading heads. Condensed water or spilled oil can prevent the correct reading in rare ocassions.*

tested it at the nominal stow position and at other arbitrary positions.

The antenna movement also worked correctly. All mesurements which involve the movement of the antenna rely on the status message, which is delivered every 200 ms. In order to visualize at a glance the parameters we used a Java client of which we show a snapshot in Fig. 5. We have also written a python client which subscribes to the notification channel and provides information on the position of the antenna (azimuth and elevation) as a function of time. Client 5 displays the most important parameters of the Status Message and plots the trajectory of the antenna in the graphical rectangle. Some of the pictures displayed in this report were taken from snapshots of that panel. There are some lines at the bottom of the client which show the status of the words using a color code. Hovering the mouse on top of the boxes triggers a pop-up message with explanatory key words.

The azimuth and elevation position is obtained from the general status in variables *azimuth encoder position* and *elevation encoder position*. Figs. 6, 7 and 8 show 3 examples of movement which depict the position, velocity and acceleration of the antenna. The maximum specified speeds are 12 degrees/sec and 6 degrees/sec in azimuth and elevation respectively. Specified acceleration is 3 degrees /sec^2 and 1 degree/sec^2 in azimuth and elevation respectively.

The three examples have been chosen using a critical distance as the separator between them. The critical distance (δ_{max}) is such that the velocity and the acceleration of the antenna acquire their maximum value and then immediately decrease. It is given by (de Vicente 2005):

$$\delta_{max} = \frac{\pi}{2} \frac{V_{max}^2}{A_{max}} \tag{1}$$

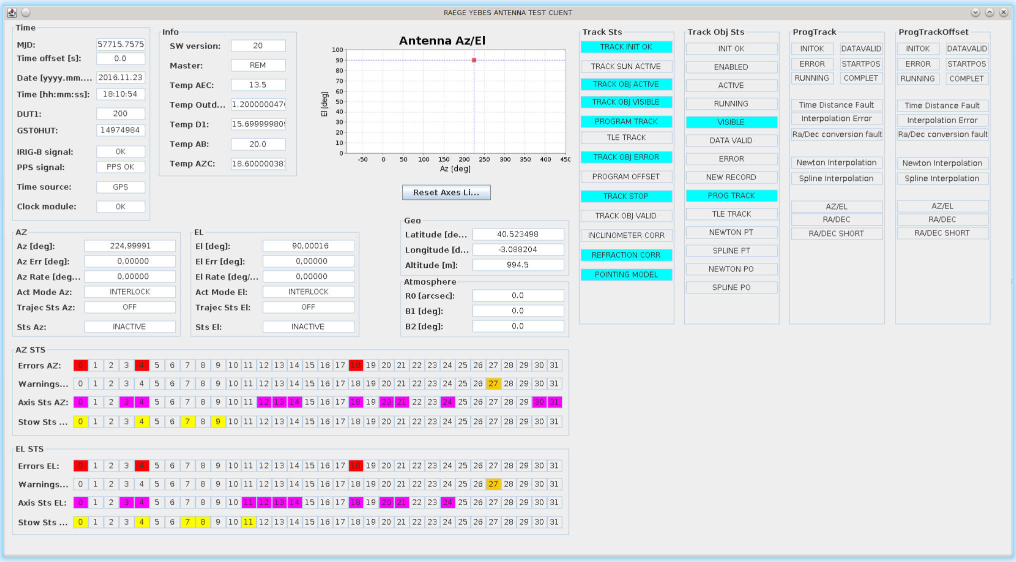

Figure 5: *Snapshot of the ACU Java client which depicts the trajectry of the antenna and a number of important variables from the status message. This client is a great diagnosis tool and allows to check the content of the variables when the mode changes.*

For the 13.2m these distances are:

$$\delta_{max}(Az) = \frac{\pi}{2}\frac{12^2}{6} = 37.70^\circ \tag{2}$$

$$\delta_{max}(El) = \frac{\pi}{2}\frac{6^2}{3} = 18.85^\circ \tag{3}$$

Measured speeds and accelerations matched the specification and are summarized in table 1.

Max. velocity Az.	Max. velocity El.	Max Acceleration Az.	Max acceleration El.
12 m/s	6 m/s	12 m/s^2	6 m/s^2

Table 1: *Maximum velocity and acceleration values*

5 Checking temperature probes

The ACU status provides temperature values from five probes installed in differente places: the azimuth cabin, the power cabinets in the control container, the control container, outside the container and close to it and in the antenna base. Fig. 9 shows a picture of some of these PT100 probes.

We have compared the outdoor probe values for 24 hours in a day with clears and clouds in late October. Results are displayed in Fig. 10. The probe yields temperature values larger than

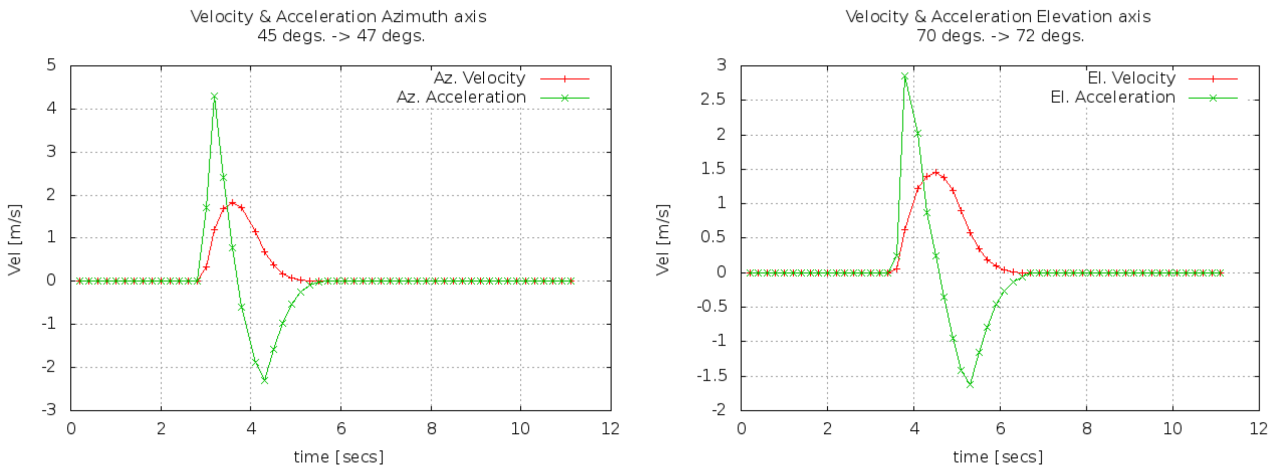

Figure 6: *Velocity and aceleration for a short length (2°) distance in azimuth and elevation. Data obtained every 200 ms from the ACU status message.*

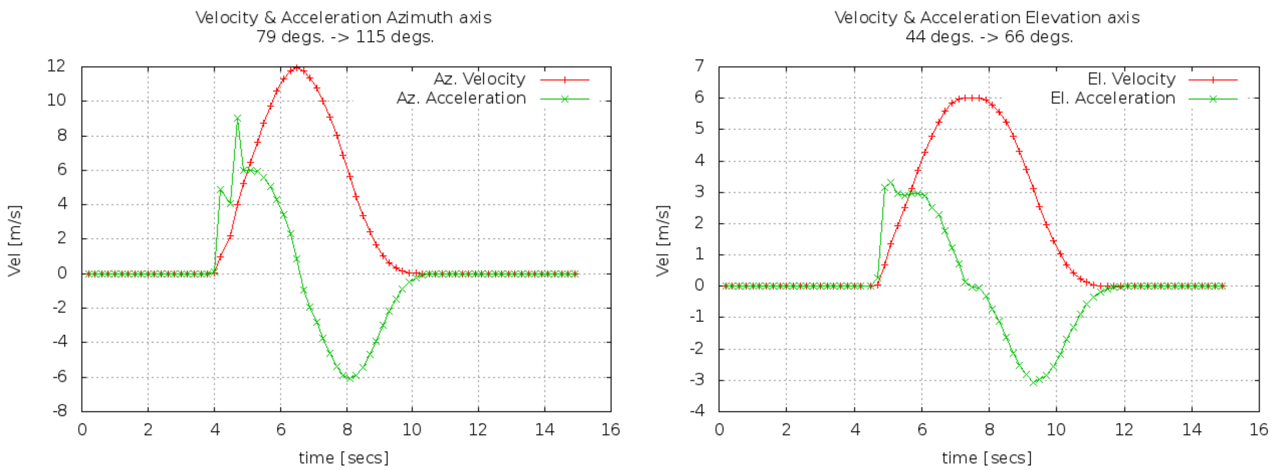

Figure 7: *Velocity and aceleration for a medium length ($\simeq 36°$ and $\simeq 20°$) movement in azimuth and elevation. Data obtained every 200 ms from the ACU status message.*

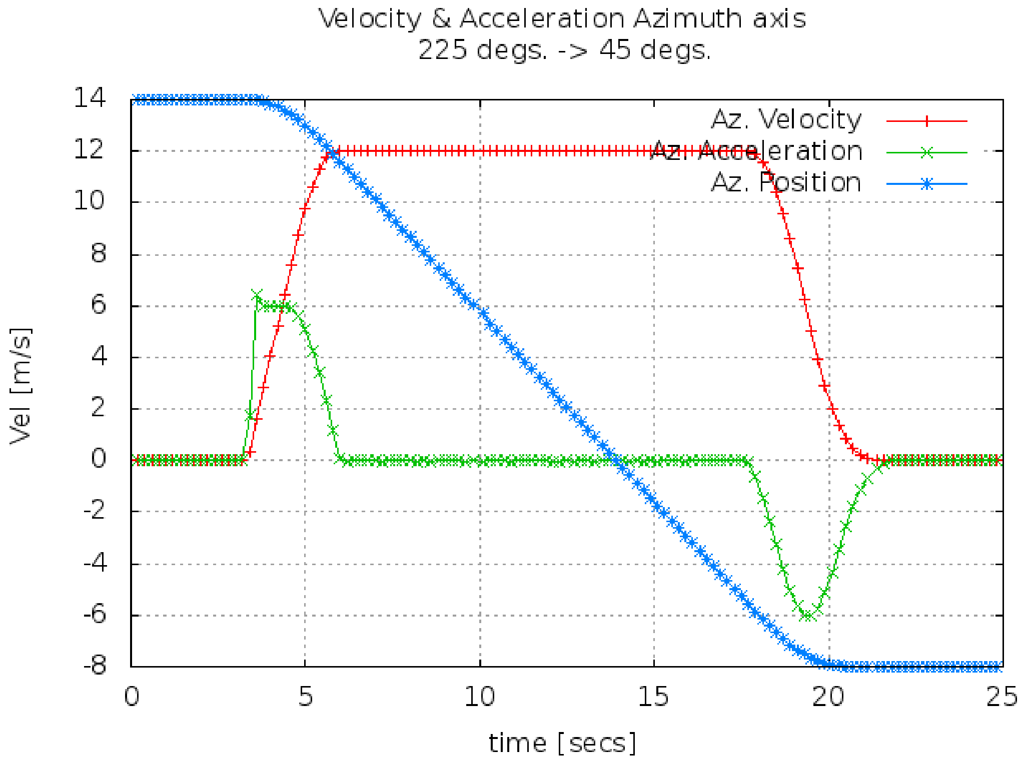

Figure 8: *Velocity and aceleration for a long length (270°) movement in azimuth and elevation. Data obtained every 200 ms from the ACU status message.*

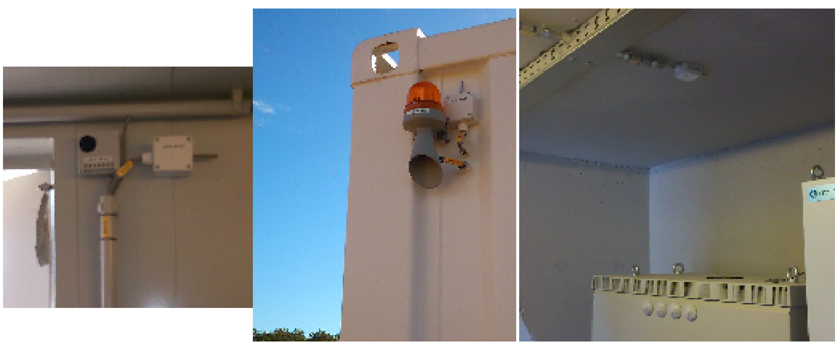

Figure 9: *MT-Mechatronics temperature probes. From left to right: inside container, outside container, antenna base*

the weather station (up to 3 degrees), but this difference is not constant and depends on the day time. The difference goes to a minimum (0.5 degrees) at night. Both, the azimuth cabin and the control container were working with the air conditioning on.

The outdoor probe is very close ($\sim$ 1 cm) to the container wall and located towards 315 degrees. The tower and the container itself cast shade on the probe up to the time when the sun is at 225 degrees azimuth. This means that the sun will illuminate directly the probe from 15:30 approximately. The proximity of the container may affect the temperature readings during the day since the wall radiates heat. These two facts probably cause the probe to generate higher values than those from the Observatory weather station.

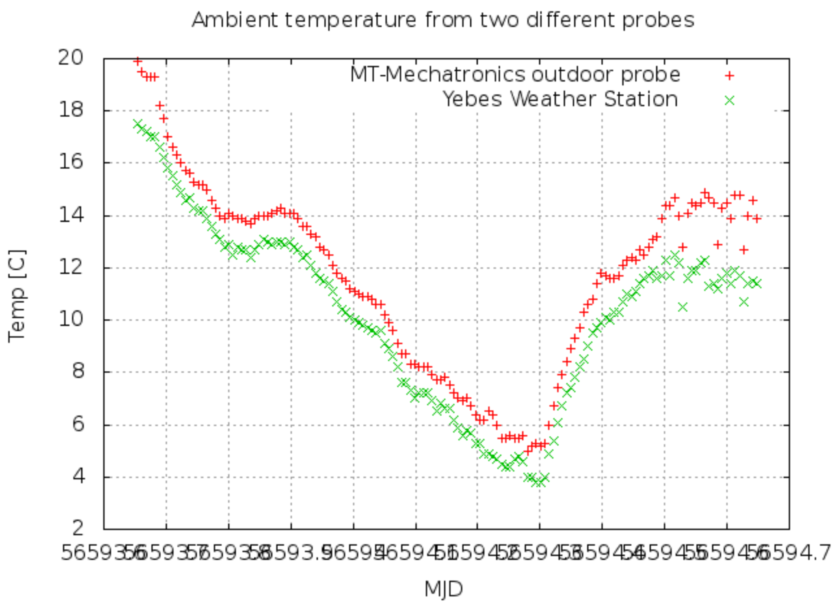

Figure 10: *Outdoor temperatures measured between October 28th and 29th during 24 hours with the weather station and the MT-Mechatronics probe.*

Indoor probes are very useful to determine if the air conditioning is working properly since high temperatures may damage or reduce the life of the equipment inside. The air conditioning system at the AEC (container) does not have a remote control: it is necessary to open the AC at its lower left side. There are two switches to adjust the AC. The air conditioning at the azimuth cabin does not have a remote and, apparently, no method to set the temperature. The azimuth cabin should also be equipped with a humidity sensor to warn the operators in case of condensation.

Recent update (year 2016): We have installed a second air conditioning system at the Container since the one installed by MT-Mechatronics was insufficient in summer and very high temperatures, up to 40 degrees, were reached. If this happens the motor amplifiers may stop working and the antenna blocks.

6 Checking the pointing model

The pointing model of the 13.2 m uses the same 7 parameter model as the one used at the 40 m radio-telescope (see de Vicente 2008). We did a very basic and partial check since only parameters P1 and P7 were checked while tracking a source.

Fig. 11 shows both cases in panels a, and b respectively. We changed parameter P1 +600 arcsecs while the source was tracking and did the same for parameter 7: we first applied a correction of +600 arcsecs and then we applied the inverse correction of -600 arcsecs, panels a and b respectively. According to Fig. 11 the antenna jumps to the new position and then it apparently goes back to the old trajectory as if the offset were cancelled. The offset was not cancelled but the reference system absorbs this change and computes azimuth and elevation in the new reference system.

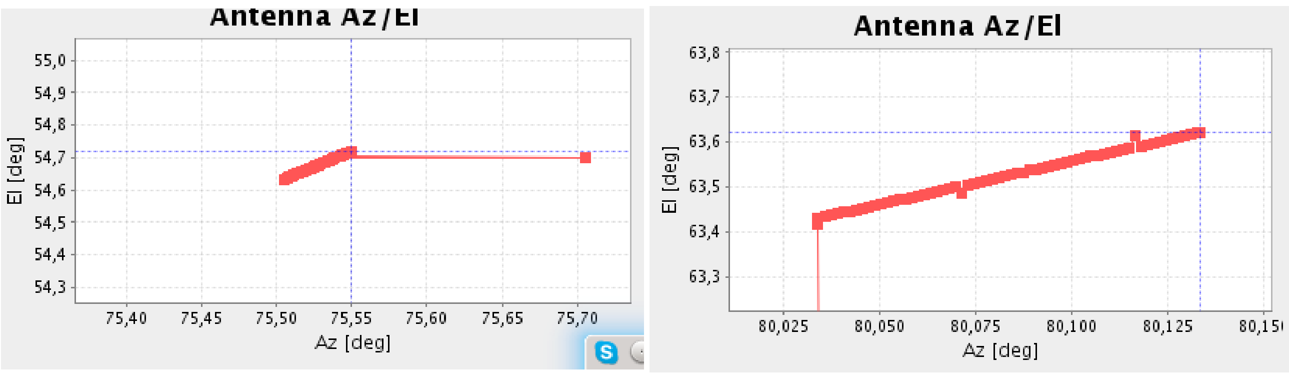

Figure 11: *Induced pointing errors: a) change in P1 by 600 arcsecs. b) Changes in P7, first -600 arcsecs and then +600 arcsecs.*

Other consistency checks were done by changing all the other P parameters one by one and keeping the others to 0 and looking how they changed at the status message. However without a frontend and a data acquisition, checking the pointing model is not easy.

2016 update: observations with a frontend and writing the data demonstrate the pointing model works as expected.

The manual pointing model was also checked issuing a manual correction of 60 arcsecs in azimuth and 120 arcsecs in elevation while tracking a source. The azimuth and elevation encoder readings jumped by the appropiate number when issuing the command. Fig. 12 shows the jump when activating the correction and when deactivating it.

7 Tracking tables

Tracking of radio sources is done loading tables in advance. Every table is composed of 3 rows: time (in UTC scale), X coordinate and Y coordinate. The X and Y coordinates can be Azimuth and Elevation, or Right Ascention and Declination. The table can hold up to a maximum of 50 elements but it is possible to append tables to the running one. The time should be supplied as Modified Julian Day and both angular coordinates in degrees. We tested the tracking with both horizontal coordinates (Az, EL) and with equatorial ones (Ra, Dec). The first test was

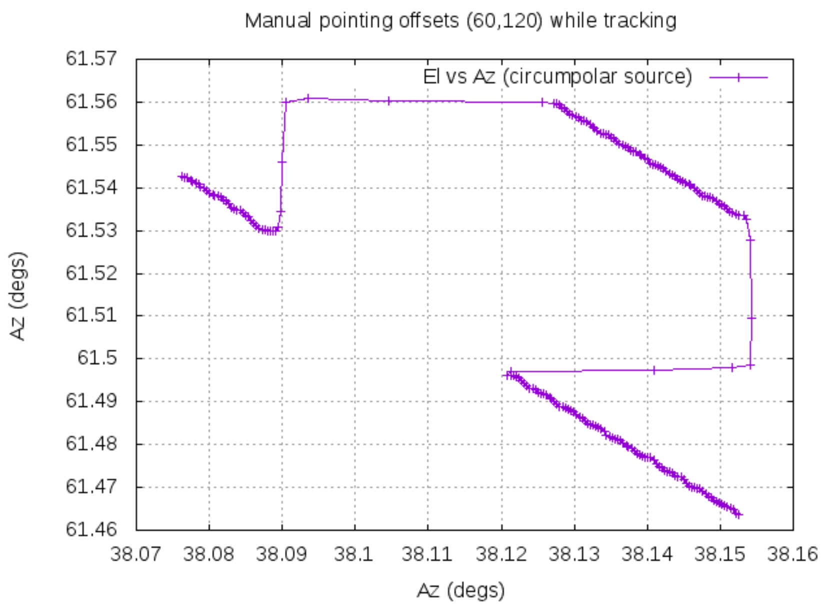

Figure 12: *Elevation versus azimuth while tracking a source. An offset of 60 arcsecs in Azimuth and 12 arcsecs in elevation was injected and later removed. Cicumpolar source moving from bottom left to top right in the plot*

unsuccesful as it can be seen in Fig. 13, due to some software issues which were corrected immediately after contacting MT-Mechatronics.

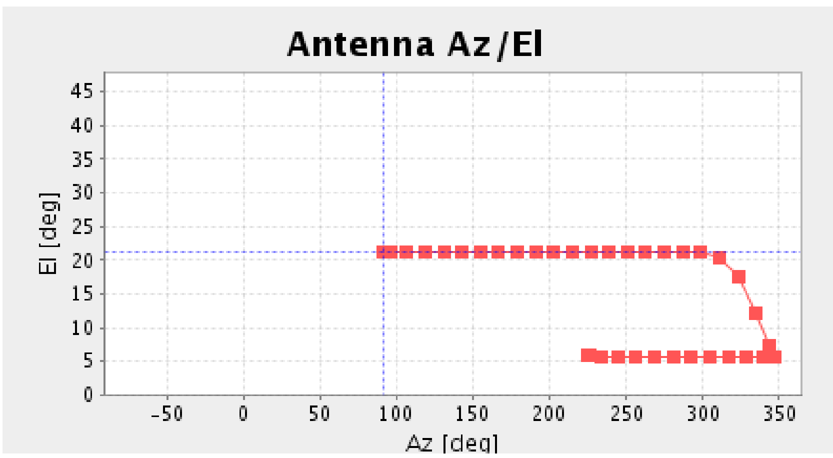

Figure 13: *Trajectory of the antenna while tracking a source with an Azimuth and Elevation table. The behaviour is totally anomalous, but it was corrected soon after this test.*

Fig. 14 shows the trajectory of the antenna in horizontal coordinates while tracking a source. This figure shows a linear trajectory with some ripples whose cause is currently unknown.

Fig. 15 depicts the position of the antenna relative to the source in horizontal and equatorial coordinates. This last figure is a measurement of the tracking error of the antenna and it basically shows that the peak to peak error is not symmetric, being 4 arcsecs in elevation and 1

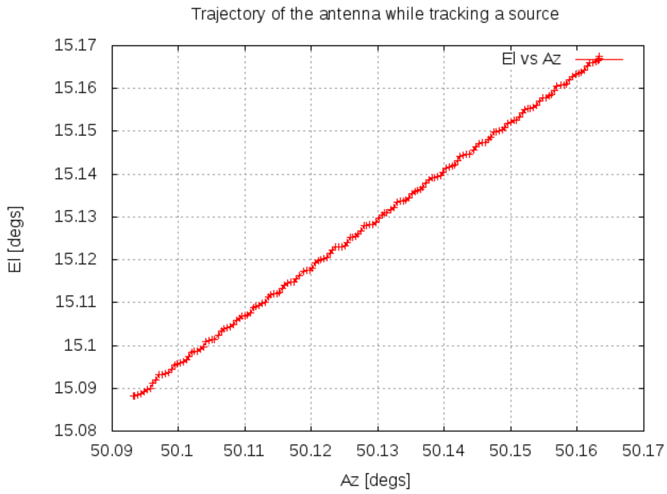

Figure 14: *Trajectory of the antenna while tracking a source with an Azimuth and Elevation table.*

arcsec in azimuth. We do not know the reason for this odd behaviour. The peak to peak error in Declination and Right Ascention is also highly non-symetric but this time the peak to peak error is the same in right ascention and declination although there is a linear dependence between them which comes from the fact that elevation errors are higher than azimuth ones. No further investigation was done to explain this behaviour since the HPBW of the telescope at 32 GHz, the highest observing frequency, is 170 arcsecs and the error is 2 % of the HPBW.

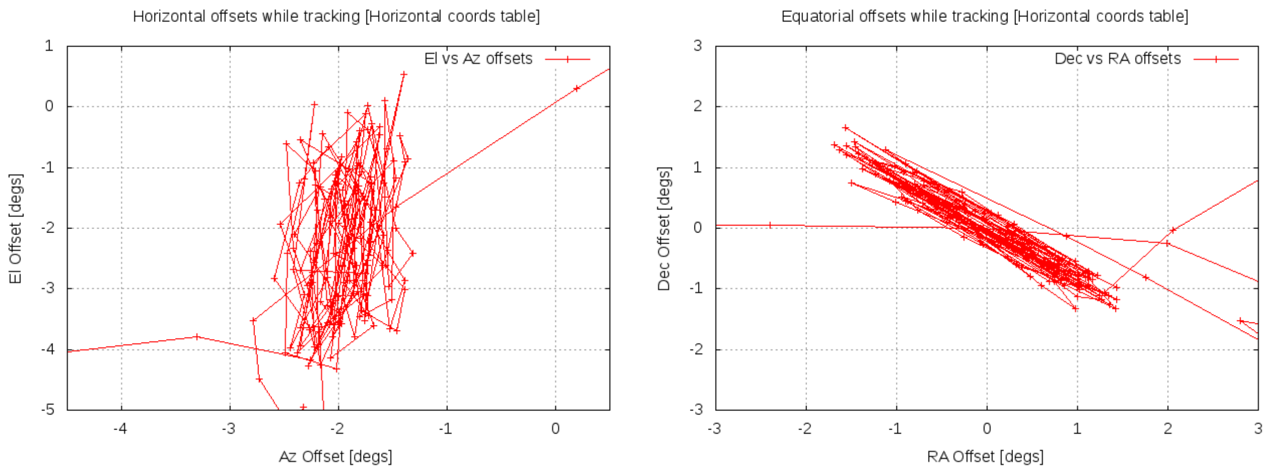

Figure 15: *Offset errors while tracking a source. a) Az, El offsets. b) Right ascention and Declination offsets*

We have also investigated if both types of tracking tables: horizontal and equatorial are consistent and yield a correct tracking trajectory. In order to check it, we generated two tables, the first one in horizontal coordinates (Az, El) and the second one in equatorial coordinates (Ra, Dec), separated by several time seconds. Fig. 16 shows both trajectories. Since both tracks lay in the same line, we can conclude that both work correctly and apparently in the same way.

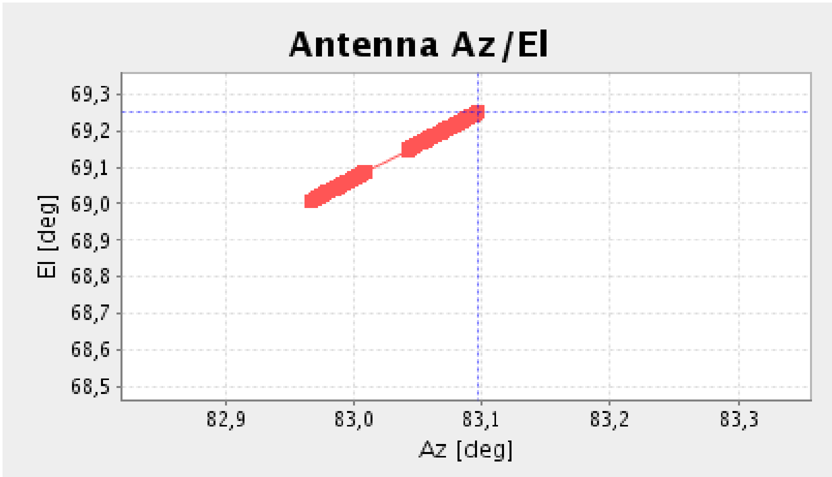

Figure 16: *Trajectory of the antenna while tracking a source with a horizontal table (Azimuth and Elevation), and later with an equatorial table (Ra and Dec).*

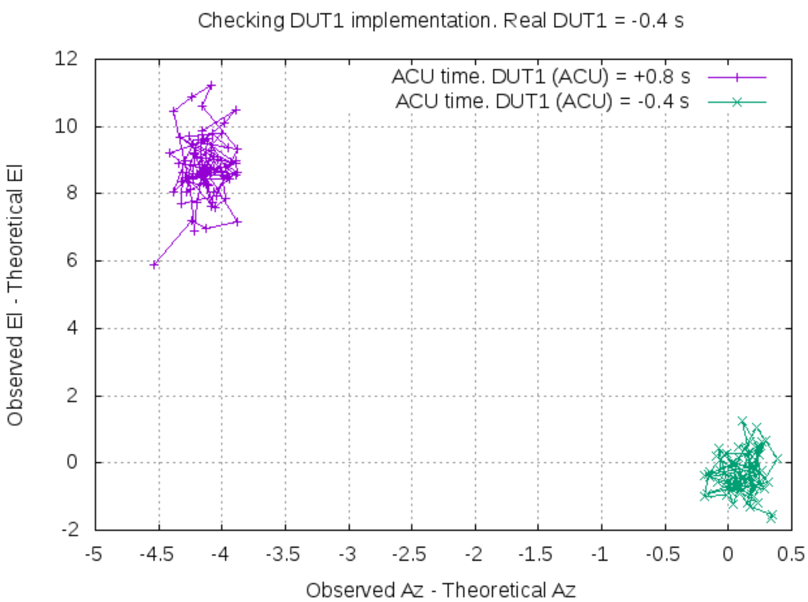

Figure 17: *Elevation error versus azimuth error while tracking a circumpolar source for several seconds in two different cases: DUT1 of +800 ms and DUT1 of -400 ms at the ACU and using the RAEGE time corrected by -400 ms. The source was moving increasing its elevation and decreasing its azimuth with time.*

In order to check if the ACU time includes the DUT1 correction for tracking when using equatorial coordinates we measured the tracking errors in two different conditions: using a DUT1 of +800 ms and -400 ms in the ACU, see Fig. 17. We computed the azimuth and elevation errors from the theoretical Az and El values using the RAEGE time system corrected by a DUT1 of -400m ms in both cases. According to Fig. 17 the errors are approximately 0 only when the RAEGE time is used and the DUT1 in the ACU is -400 ms, whereas when using a DUT1 of 800 ms the errors are close to 10 arcsecs and are due to the artificial time difference introduced by the DUT1 at the ACU. The trajectory of the cirumpolar source in an Elevation versus Azimuth diagram was from bottom left to upper right, that is increasing elevation and decreasing azimuth with time, thus the shift from the bottom right corner to the left upper corner in Fig. 17.

We conclude that the DUT1 correction is correctly applied when using equatorial tracking tables according to the algorithms in de Vicente (2005).

8 Offset tracking table

The 13.2 m ACU allows to command an offset table which depends on time and which can be added on top of the main tracking table. The offsets can be selected to be horizontal ones (azimuth in the sky and elevation) or equatorial (Right Ascention and Declination). We have tested the behaviour of the offsets table in both modes.

We checked the behaviour of the offset table in azimuth and elevation while tracking two radio sources at different elevations. The results are displayed in Fig. 18. The goal of the test was to check if the offsets, when used in Azimuth and Elevation mode, correspond to colimation (azimuth in the sky) or to encoder azimuth offsets, and to verify if the offsets are the same in both axes. Fig. 18 shows the pointing drift on a source at 24 degrees elevation (panel a) and at 52 degrees elevation (panel b). The arms that compose the cross, are not perpendicular since the antenna is tracking the source, and they do not have the same length. Moreover the length of the azimuth arm is larger at higher elevations as expected:

$$\delta Az = \delta AzS / \cos(el)$$

Fig. 19 shows both drifts simultaneously on the reference system of the radiosource being tracked. Both arms have the same length for the two elevation cases since the azimuth offsets are not encoder offsets but offsets on the sky.

Therefore we can conclude that offset tables work correctly in horizontal coordinates and the azimuth offset is a collimation offset and not an encoder offset, and MT-Mechatronicsi complied with our request.

9 HXP elevation offset

MT-Mechatronics has determined an offset versus elevation curve for the hexapod based on a finite element model of the antenna. Four axes require a modification to keep the hexapod in its

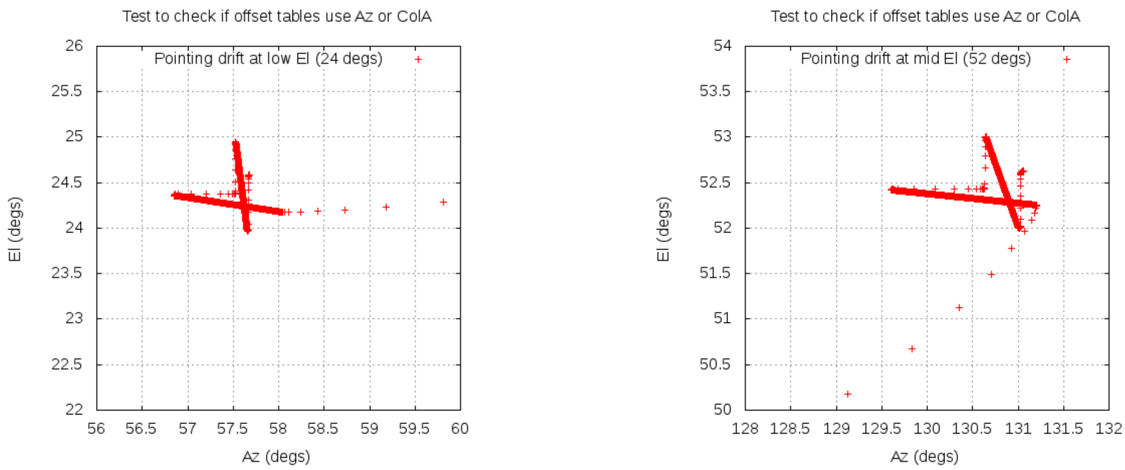

Figure 18: *Pointing drifts in azimuth and elevation at 24 and 52 degrees elevation.*

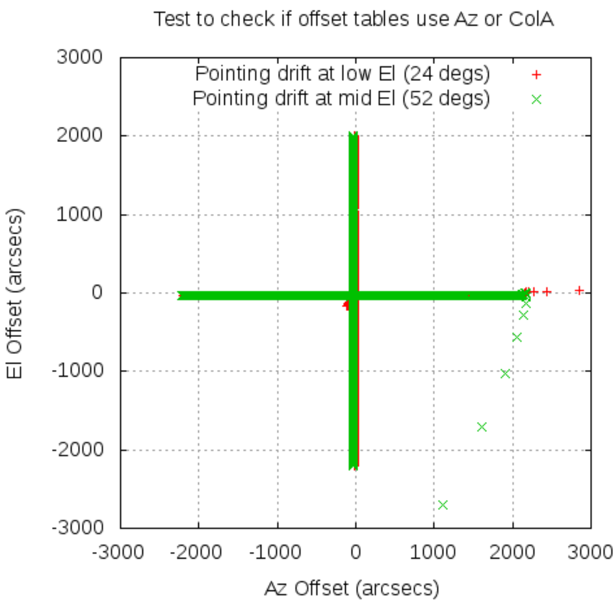

Figure 19: *Pointing drifts in azimuth and elevation at 24 and 52 degrees elevation in the reference system of the source.*

position relative to the primary focus. We have monitored the position of the subreflector as a function of elevation to check if this model is correctly applied. Fig. 20 shows the result. The correction is apparently done as required. The theoretical values were set by the author using the following expressions to determine the offsets:

$$X = 0.002 - 0.00006\, el \tag{4}$$

$$Y = 7 - 9\,\cos(el) \tag{5}$$

$$Z = 1.2 - 2\,\sin(el) \tag{6}$$

$$TX = 540 - 720\,\cos(el) \tag{7}$$

where el is the elevation of the antenna. The units for axes X, Y and Z are mm and arcsecs for TX

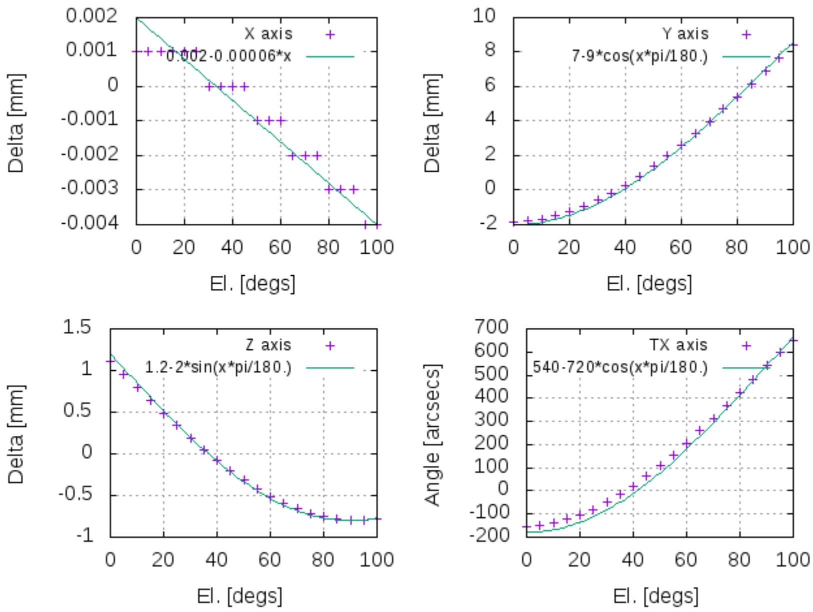

Figure 20: *Observed (red crosses) and theoretical (green line) position of the subreflector along four different axes as a function of elevation.*

10 HXP program offset

The subreflector movement ihas six degrees of freedom and it is peformed by an hexapod but the six actuators of the system do not lay along the axes the observer are used to. The system however translates from the physical axes to the common ones: lateral displacement in X and Y, displacement along the axis symmetry Z, and tilts around the axes, X, Y and Z.

The subreflector can move as a function of time by providing a table with lines which contain the time, and the position of the six axes. This feature is crucial to be able to perform focus scans in which the subreflector is usually moved along the Z axis. We tested the movement of the subreflector for the Z axis, by moving it between -10 and +10 mm. The result is shown in

Fig. 21 where we show the position of the axes Xm Y, Z, TX, and TY as a function of time. The movement is only seen for the Z axis as expected. The commanded speed is 1.2 mm/s.

Using this type of observations and analysis we have checked the movement of the subreflector along the different axes. The speed for slewing between positions along linear axes (X, Y and Z) is 2 mm/s, whereas for tilts is 240 arcsecs/s.

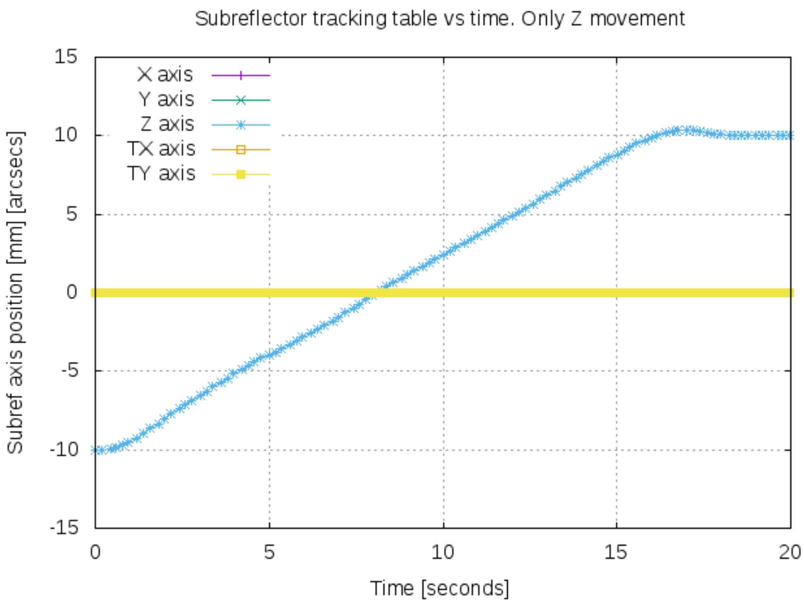

Figure 21: *Movement along the Z axis as a function of time*

11 About the implementation of ACU remote commands

This section holds some notes about the implementation of some ACU commands at the high level antenna component that we considered interesting during the checking phase. Throught this section we mention functions that belong to the raegeAntenna component and to highlight them we use italics. The reader should consider them a sort of "pseudocode" since no calling parameters are specified in any case for the sake of clarity.

Azimuth and Elevation axes are considered to be two different subystems, as the subreflector or the tracking mode. In practice this means that commanding movements per axis is done with two different commands instead of a single one which uses two different parameters to specify the azimuth and elevation properties, like target position or velocity for each encoder. For example, unstowing the antenna requires commands *unstowAz()* and *unstowEl()*, because they talk to different subsystems.

Tracking commands require a composition of four commands in this order:

- *trackSetup()* Sets the tracking functionality and allows to use a NORAD TLE two line elements or a program track table.

- *loadProgramTrackTable()* Loads the desired program track table. For satellites NORAD function *loadNORAD()* should be used instead.
- *programTrackAz()* "Connects" the output from the program table to the Azimuth axis, "feeding" it with that table.
- *programTrackEl()* "Connects" the output from the program table to the Elevation axis, "feeding" it with that table.

If an offset table has to be used on top of the main tracking table we usually load it immediately after the previous sequence:

- *loadProgramTrackOffsetTable()*

Tables can be loaded immediately or according to a schedule. We usually load them according to a schedule. For that we compute previously the time required for slewing plus some overheads and add this difference to the current time. In special ocassions like for tracking satellites we immediately load the table. If a table is immediately loaded and the initial time has not ellapsed, the telescope will move to the first position of the table and stay there until the time arrives, at which time it will follow the interpolated trajectory. If there is an offset table the antenna will move to the first element of that table.

Loading new tables does not require stopping the old one by loading a new table with Load Mode Keyword *Reset*. It just requires loading a new table with Load Mode Keyword *New*. This is specially interesting for VLBI observations because it reduces by a great ammount the time required to load new observations. The philosophy with the 40m was to cancel the old table by sending a new one with *Reset*, stopping the antenna later and loading a new table. This is no longer necessary and saves about 10 seconds of time.

Moving the subreflector according to tables, that is versus elevation or versus time, requires a policy similar to tracking tables. For movements which depend on elevation we follow this sequence:

- *stopMotor(hexapod)* Stops the hexapod motors (6 of them).
- *loadHexapodElevationOffsetTable()* Load the dependency with elevation for the 6 axes.
- *hexapodElevationTable()* Activate the corrections for the hexapod
- *activateHxp()* Activate the hexapod motors
- *programTrackHxp()* "Connects" the output from the elevation table to the subreflector subsystem, "feeding" it with that table.

This same sequence can be used to deactivate any dependency and it can be done either setting *hexapodElevationTable()* to use OFF as parameter or loading a zero offset non dependent elevation table with command *loadHexapodElevationOffsetTable()*

Manual pointing corrections are achieved using commands *trackingOffsetAz()* and *trackingOffsetEl()* since they work as offsets in the sky and not as offsets for the encoders and are only appliable when tracking is being done (like the DUT1 implementation). These commands should not be confused with *positionOffsetAz()* and *positionOffsetEl()* which inject offsets in the azimuth and elevation encoders respectively at any moment. These latter commands are not used in high level procedures at the control system.

Azimuth and Elevation values at the high level antenna component are obtained from the status message using *encoderAz()* and *encoderEl()* (variables *axisStatusACU[].encoderPosition*) and substracting any current pointing offset caused by the above commands (manual pointing corrections). These offsets can be recovered from the *trackingStatus* section and its variables are: *trackingStatus.posOffsetAz* and *trackingStatus.posOffsetEl*. Once this substraction is performed we use a new horizontal system which has already absorbed these changes. This is an importan requirement for the data acquisition and tagging of data together with horizontal coordinates.

Corrections by the pointing model and the refraction are already in the encoder horizontal system but they are only applied when the tracking mode is activated. In this case the system absorbs these corrections and there is no way know them except by looking up at variables *trackingStatus.pointingModel [9]*, *trackingStatus.pointingModelCorrectionAz*, *trackingStatus.pointingModelCorrectionEl* and *trackingStatus.refractionModelCorrection* in the status message.

References

[1] P. de Vicente *Checking the pointing corrections implementation for the 40m radiotelescope*, IT-CDT 2008-7

[2] P. de Vicente *Movimientos de tránsito y seguimiento de planetas con el radiotelescopio de 40m*, IT OAN 2005-11

[3] P. de Vicente, M. Visus *Tracking errors at the 40 m radiotelescope*, IT-CDT 2013-13