Interfacing two ALMA softwares: TELCAL within CASA

Jean-Christophe Roche (roche@iram.fr), Institut de Radio Astronomie Millimétrique
Robert Lucas, European Southern Observatory
Dominique Broguiere, Institut de Radio Astronomie Millimétrique
Juan R. Pardo-Carrión, Consejo Superior de Investigaciones Científicas

Context

The Atacama Large Millimeter Array (**ALMA**) is a radio interferometer of 66 antennas that is under construction and commissioning at the Chajnantor plateau at 5000 meters altitude in the Atacama desert in northern Chile.

Telescope calibration consists in a series of operations (adjustment of antennas, atmospheric calibrations and early evaluation of the quality of observations) that are performed in order to ensure that the ALMA telescope is and remains in proper shape to successfully observe the projects that are scheduled. This key step in the observing process is done by **TELCAL** on-line software.

CASA (Common Astronomy Software Applications) is the off-line data reduction software (final calibration, imaging and deconvolution) chosen for ALMA.

Aims

TELCAL main purpose is to execute calibrations at the telescope in quasi-realtime, i.e. on-line, with automatic feedback to operators. In addition TELCAL has to be available to perform the same calibrations off-line, as a tool for data quality control and commissioning. For this purpose it was decided to interface TELCAL within CASA.

This poster first (1) explains how to interface the two ALMA softwares. Then (2) it describes the way to automatically extract the needed code from the ACS (ALMA Common Software) development environment to create a TELCAL standalone package, that is to say without any ACS dependency. And (3) the list of TELCAL tasks, which are executable within CASA, is given. Finally (4) an example of executing a TELCAL task is proposed.

References

ALMA	*http://www.almaobservatory.org*
CASA	*http://casa.nrao.edu*
IRAM	*http://www.iram-institute.org*
snakefood	*http://furius.ca/snakefood*
SWIG	*http://www.swig.org*

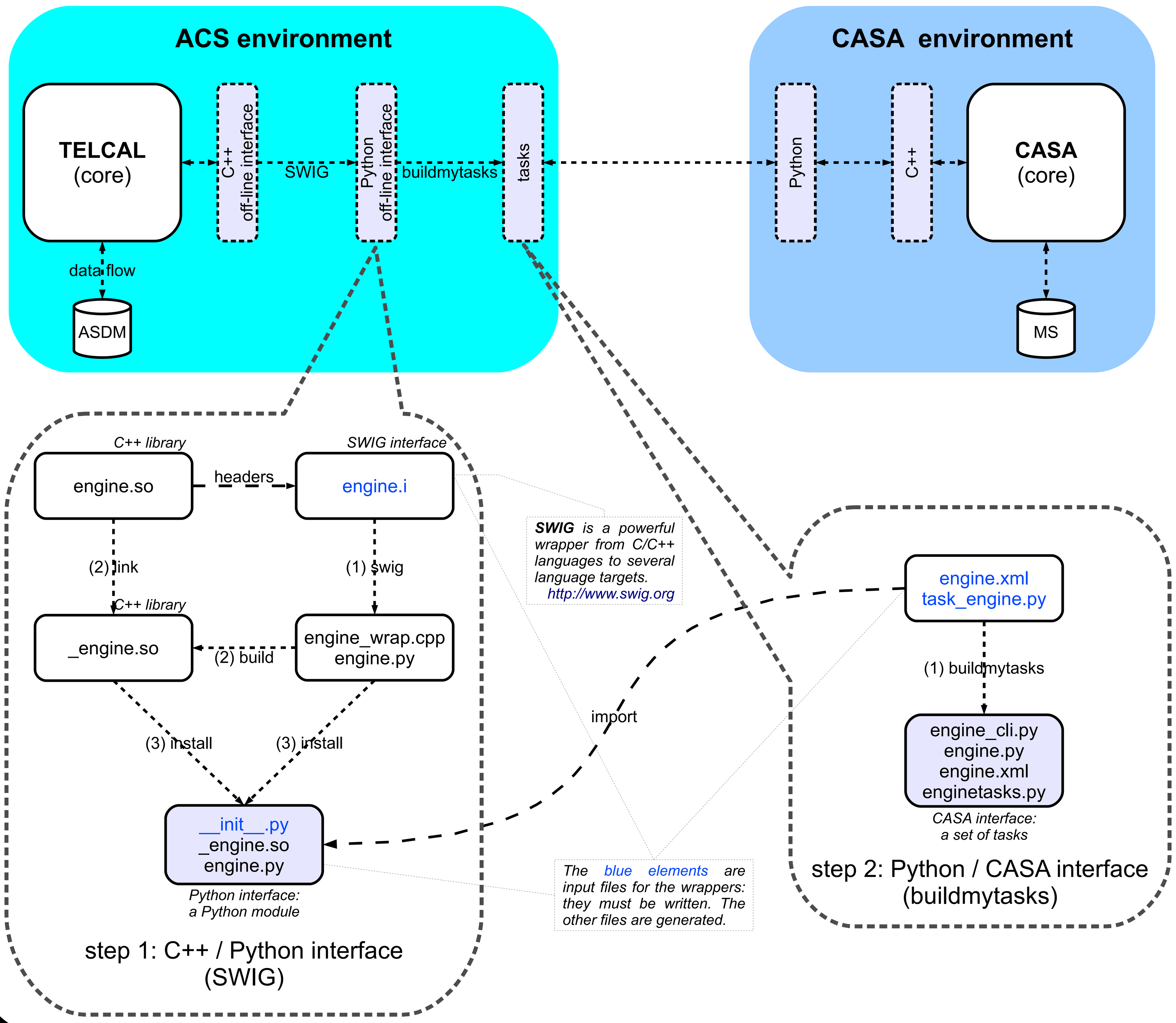
Acronyms

ACS	ALMA Common Software
ALMA	Atacama Large Millimeter Array
ASDM	ALMA Science Data Model
CASA	Common Astronomy Software Applications
MS	Measurement Set
NRAO	National Radio Astronomy Observatory
SWIG	Simplified Wrapper and Interface Generator

1 Create tasks: two interfacing steps

TELCAL is the on-line calibration software.
TELCAL has been designed and is continuously developed at IRAM since 2002. Around 70% of the 10^5 lines of code is written in C++. It is developed in the ACS environment (CORBA based middleware).
Both TELCAL input data and output results are formulated according the ASDM (ALMA Science Data Model).

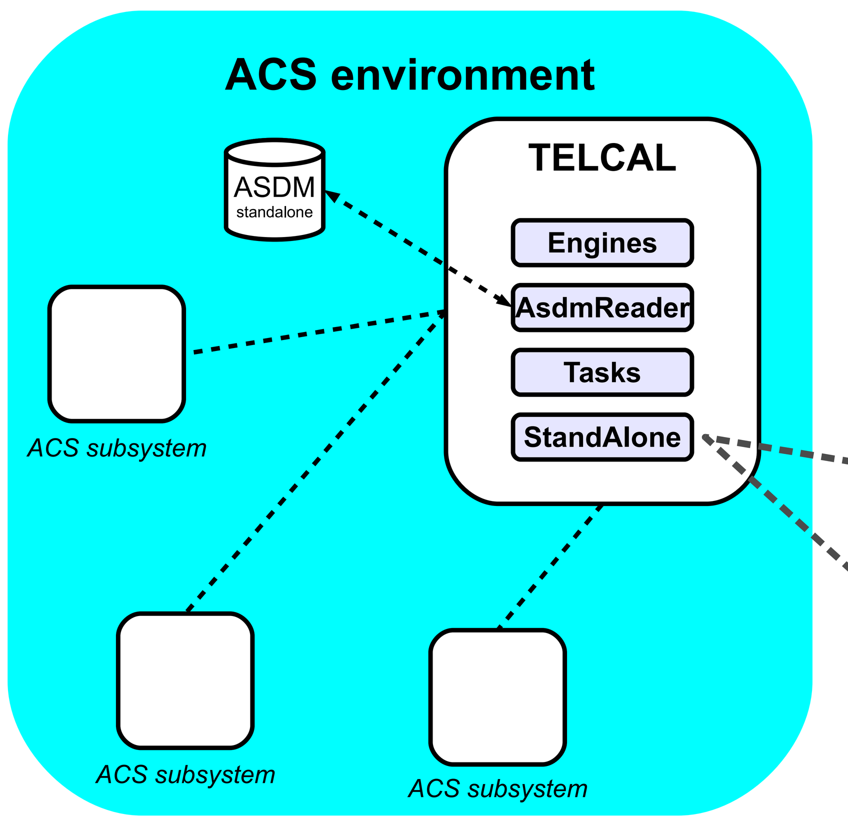
CASA is the off-line data reduction software.
CASA is developed at NRAO. It consists of a set of C++ tools bundled together under a set of **tasks** available through the IPython interactive shell that offers enhanced introspection, additional shell syntax, syntax highlighting, tab completion and rich history.
CASA accesses data through the MS (Measurement Set) data model.

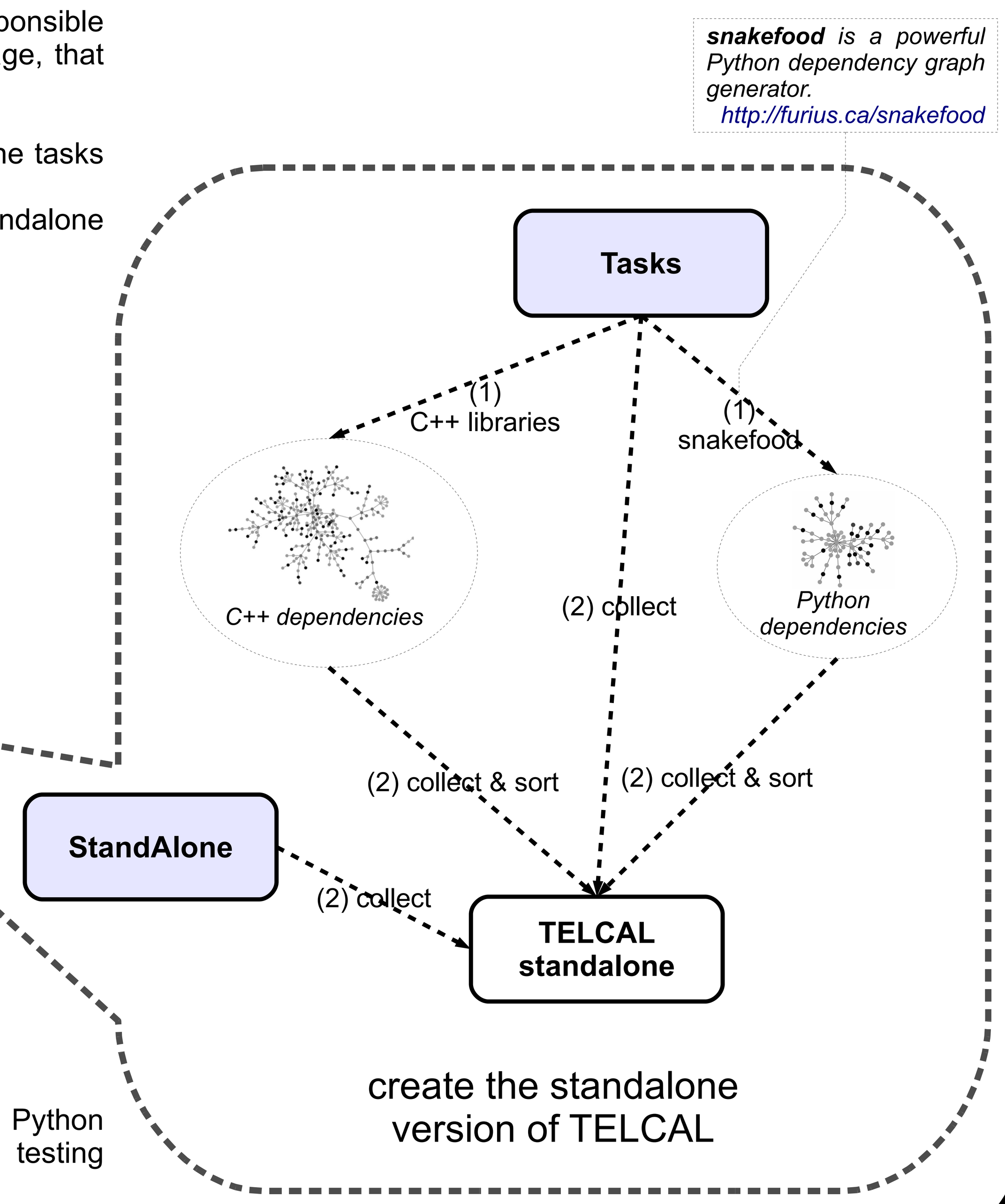


2 The StandAlone module

The **StandAlone** module is a part of the TELCAL subsystem responsible to create a packaging of TELCAL, the TELCAL standalone package, that works outside of the ACS environment.
It builds the TELCAL standalone package in two steps:

- (1) resolve all C++ and Python dependencies that belong to the tasks (~**3670** dependencies);
- (2) collect and sort those dependencies to create the standalone package (**~330** files).





The **TELCAL standalone** package is a set of C++ libraries, Python modules and package files (standalone installation, information, testing environment, ...) extracted for the TELCAL subsystem.

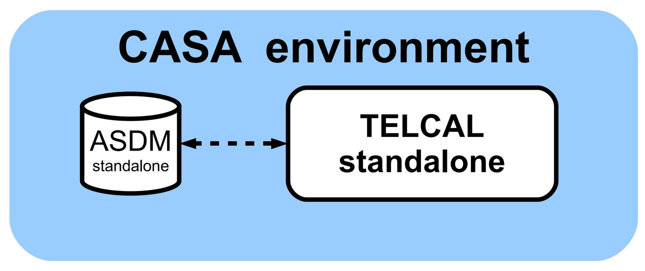
3 TELCAL tasks within CASA

A **TELCAL task** is tuned with several parameters that depend on the calibration engine. The following table shows the current list of tasks and their parameters.

task	short description	asdm	dataorigin	scans	antennas	calresult	showplot	verbose	specific parameters
asdmlist	List the scan and subscan of an ASDM dataset.	x						x	subscans
asdmplot	Plot the contents of an ASDM dataset.	x		x	x			x	xaxis yaxis bbpairs polarizations phasecorrections amplog
tc_amplical	Process an amplitude calibration on an ASDM and plot the results.	x	x	x		x	x	x	
tc_amplicurve	Process an amplicurve calibration on an ASDM and plot the results.	x	x	x		x	x	x	polydegree
tc_antpos	Process a antenna position calibration on an ASDM and plot the results for a given antenna.	x		x		x	x	x	antenna ignoredscans mode
tc_atmosphere	Process an atmosphere calibration on an ASDM and plot the results.	x	x	x	x	x	x	x	trecmode
tc_bandpass	Process a bandpass calibration on an ASDM and plot the results.	x		x		x	x	x	bbpairs
tc_delay	Process a delay calibration on an ASDM and plot the results.	x		x	x	x	x	x	bbpairs
tc_focus	Process a focus calibration on an ASDM and plot the results.	x	x	x	x	x	x	x	axis fitwidth
tc_phasecal	Process a phase calibration on an ASDM and plot the results.	x	x	x		x	x	x	
tc_phasecurve	Process an phasecurve calibration on an ASDM and plot the results.	x	x	x		x	x	x	polydegree
tc_pointing	Process a pointing calibration on an ASDM and plot the results.	x	x	x	x	x	x	x	fitwidth
tc_wvr	Plot the phase in an ASDM dataset with WVR correction.	x		x	x	x	x	x	

4 TELCAL-CASA interface at work

The **TELCAL standalone** package is now ready to be executed inside the CASA environment. This panel describes how the TELCAL tasks work through an example: the **tc_pointing** task that performs a TELCAL pointing calibration in CASA.



The two first steps are:

- ❶ start CASA;
- ❷ load the TELCAL standalone package.

❶
```
> casapy
CASA Version 3.1.0 (r12727)
  Compiled on: Wed 2010/09/15 06:41:24 UTC
____________________________________________________________________
    For help use the following commands:
    tasklist               - Task list organized by category
    taskhelp               - One line summary of available tasks
    help taskname          - Full help for task
    toolhelp               - One line summary of available tools
    help par.parametername - Full help for parameter name
    Single Dish sd* tasks are available after asap_init() is run
____________________________________________________________________
Activating auto-logging. Current session state plus future input saved.
Filename       : ipython.log
Mode           : backup
Output logging : False
Raw input log  : False
Timestamping   : False
State          : active
CASA <2>:
```

❷
```
CASA <2>: execfile('telcaltasks.py')

CASA <3>:
```

❷
```
Time                Priority  Origin        Message
2010-11-04 13:55…   INFO                    ::casa
2010-11-04 13:56…   INFO                    ::casa
2010-11-04 13:57…   INFO      casa::::casa  ...
2010-11-04 13:57…   INFO      casa::::casa  CASA Version 3.0.0 (build #9860)
2010-11-04 13:57…   INFO      casa::::casa    Built on: Sat 2009/12/19 04:59:07 UTC
2010-11-04 13:57…   INFO      casa::::casa  loading asdmlist task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading asdmplot task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_amplical task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_amplicurve task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_antpos task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_atmosphere task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_bandpass task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_delay task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_focus task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_phasecal task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_phasecurve task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_pointing task from TelCal package ...
2010-11-04 14:01…   INFO      casa::::casa  loading tc_wvr task from TelCal package ...
```

CASA logs

Then consider the **tc_pointing** task which processes a pointing calibration on an ASDM:

- ❸ get help on the task;
- ❹ list the parameter default values of the task.

❸
```
CASA <3>: help tc_pointing

CASA <4>:
```

❸
```
Help on tc_pointing task:

Process a pointing calibration on an ASDM and plot the results.
  tc_pointing task:

  Process a pointing calibration on an ASDM and plot the results.

  Keyword arguments:

  asdm -- name of input ASDM dataset
  default: ''
  example: 'testPointingModel'

  dataorigin -- origin of the data to be used
  default: 'avercross'
  available: 'avercross'  - channel averaged cross-correlation data
             'averauto'   - channel averaged auto-correlation data
             'totalpower' - total power from broad-band detectors

  scans -- scans to be plotted
  default: '' = all scans
  example: '3701'

  antennas -- list of antennas
  default: '' = take from the ASDM dataset
  example: 'DV01 DV02'

  fitwidth -- fit beam width
  default: True

  calresult -- name of output ASDM dataset
  default: asdm + '_pointing_result'
  example: 'testPhase_pointing_result'

  showplot -- whether or not display the plot
  default: True

  verbose -- verbose mode
  default: False
```

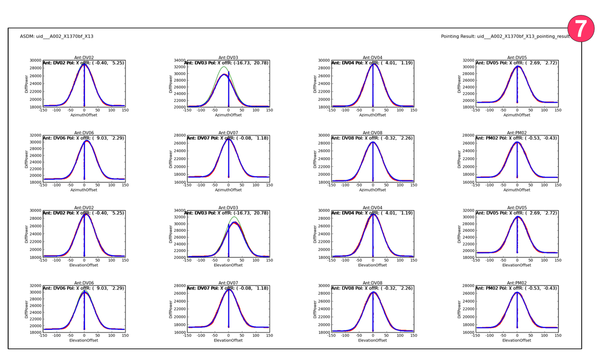
❹
```
CASA <4>: inp tc_pointing
--------> inp tc_pointing
#  tc_pointing :: Process a pointing calibration on an ASDM and plot the results.
asdm           =         ''        #  name of input ASDM dataset
dataorigin     = 'avercross'       #  origin of the data to be used
scans          =         ''        #  scans to be plotted
antennas       =         ''        #  list of antennas
fitwidth       =       True        #  fit beam width
calresult      =         ''        #  name of output ASDM dataset
showplot       =       True        #  whether or not display the plot
verbose        =      False        #  verbose mode
async          =      False        #  If true the taskname must be started using tc_pointing(...)

CASA <5>:
```

Finally tune the parameters, execute the task and read the results of the calibration, then view the plot generated by the task:

- ❺ tune the parameters;
- ❻ execute the task and read the calibration results;
- ❼ view the plot.

❺
```
CASA <5>: asdm='uid___A002_X1370bf_X13'

CASA <6>: dataorigin='totalpower'

CASA <7>: scans='1'

CASA <8>: inp(tc_pointing)
#  tc_pointing :: Process a pointing calibration on an ASDM and plot the results.
asdm           = 'uid___A002_X1370bf_X13' #  name of input ASDM dataset
dataorigin     = 'totalpower'      #  origin of the data to be used
scans          =        '1'        #  scans to be plotted
antennas       =         ''        #  list of antennas
fitwidth       =       True        #  fit beam width
calresult      =         ''        #  name of output ASDM dataset
showplot       =       True        #  whether or not display the plot
verbose        =      False        #  verbose mode
async          =      False        #  If true the taskname must be started using tc_pointing(...)

CASA <9>:
```

❻
```
CASA <9>: go
---------> go()
Executing:  tc_pointing()
```

❼


❻
```
2010-11-04 14:07… INFO                        tc_pointing::::casa
2010-11-04 14:07… INFO   tc_pointing::::casa  ##########################################
2010-11-04 14:07… INFO   tc_pointing::::casa  ##### Begin Task: tc_pointing       #####
2010-11-04 14:07… INFO                        tc_pointing::::casa
2010-11-04 14:07… INFO   tc_pointing::::casa  asdm       : uid___A002_X1370bf_X13
2010-11-04 14:07… INFO   tc_pointing::::casa  dataorigin : totalpower
2010-11-04 14:07… INFO   tc_pointing::::casa  scans      : 1
2010-11-04 14:07… INFO   tc_pointing::::casa  antennas   :
2010-11-04 14:07… INFO   tc_pointing::::casa  fitwidth   : True
2010-11-04 14:07… INFO   tc_pointing::::casa  result     : uid___A002_X1370bf_X13_pointing_result
2010-11-04 14:07… INFO   tc_pointing::::casa  showplot   : True
2010-11-04 14:07… INFO   tc_pointing::::casa  verbose    : False
2010-11-04 14:07… INFO   tc_pointing::::casa  Start: 2010-09-30T23:33:53.136000000 End: 2010-10-01T00:33:53.136000000 Duration 3600.000s
2010-11-04 14:07… INFO   tc_pointing::::casa  Method: CROSS Band: ALMA_RB_01 Freq:  86.13- 86.13GHz Temp: 264.75 Az  81.17 El  26.15
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV02  Pol: Y offRel: ( -0.40,   5.25) offAbs: (158.46, -52.78) Err:(0.05, 0.05) Peak: 10787.076252 ( 10.51) Width:  75.44 (0.09)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV03  Pol: Y offRel: (-16.73,  20.78) offAbs: (-604.52, 475.05) Err:(0.08, 0.07) Peak: 11860.345144 ( 26.47) Width:  74.45 (0.16)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV04  Pol: Y offRel: (  4.01,   1.19) offAbs: (505.20,  44.50) Err:(0.05, 0.05) Peak: 10883.165652 ( 11.28) Width:  74.87 (0.10)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV05  Pol: Y offRel: (  2.69,   2.72) offAbs: (-253.90, 181.23) Err:(0.05, 0.05) Peak: 10881.604850 ( 10.93) Width:  74.77 (0.10)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV06  Pol: Y offRel: (  9.03,   2.29) offAbs: (-312.77, -90.58) Err:(0.04, 0.04) Peak: 11600.654289 ( 10.80) Width:  72.20 (0.08)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV07  Pol: Y offRel: ( -0.08,   1.18) offAbs: (388.89, -97.57) Err:(0.05, 0.05) Peak: 9718.288530 (  9.66) Width:  72.74 (0.09)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: DV08  Pol: Y offRel: ( -0.32,   2.26) offAbs: (-299.12, -341.39) Err:(0.05, 0.05) Peak: 9982.332259 (  9.72) Width:  76.35 (0.10)
2010-11-04 14:07… INFO   tc_pointing::::casa  Ant: PM02  Pol: Y offRel: ( -0.53,  -0.43) offAbs: ( 55.81, 380.04) Err:(0.05, 0.05) Peak: 9081.419037 (  8.76) Width:  75.19 (0.09)
2010-11-04 14:08… INFO                        tc_pointing::::casa
2010-11-04 14:08… INFO   tc_pointing::::casa  ##### End Task: tc_pointing         #####
2010-11-04 14:08… INFO   tc_pointing::::casa  ##########################################
```

CASA logs