
P-AIRCARS

Release 1.0.0

Devojyoti Kansabanik

Oct 05, 2022

P-AIRCARS

1	Introduction	3
1.1	P-AIRCARS (Polarimetry using Automated Imaging Routine for Compact Arrays of the Radio Sun) .	3
1.2	System requirements	4
1.2.1	Minimum hardware requirements	4
1.2.2	Workstation configuration	4
1.2.3	Software environment	4
2	Installation	5
2.1	Creating virtual environment	5
2.2	Obtaining P-AIRCARS	5
2.3	Installing P-AIRCARS	5
3	Tutorial	7
3.1	Start the P-AIRCARS	7
3.2	Download the data	8
3.3	Fill up the inputs	9
3.3.1	Manual input	9
3.3.2	Load from previous input file	9
3.4	Validating inputs	9
3.5	Save inputs	9
3.6	Run the P-AIRCARS	10
3.7	Notification over e-mail	10
3.8	Locate the final images	10
4	Details of Inputs	11
4.1	Inputs	11
4.2	Advanced Inputs	13
4.3	High Performance Computing (HPC) settings	13
5	Robustness and Quality	15
5.1	Quality (Q)	15
5.2	Robustness (R)	15
5.3	Combinations	15
6	paircars	19
6.1	paircars	19
6.1.1	paircars.access_ms	19
6.1.2	paircars.basic_func	19
6.1.3	paircars.decor	19
6.1.4	paircars.dynamic_spectrum	19
6.1.5	paircars.flagger	19

6.1.6	paircars.fullpol_selfcal_LTS	19
6.1.7	paircars.intensity_selfcal_LTS	19
7	paircars scripts	21
7.1	paircars scripts	21
7.1.1	control_pairedcars	21
7.1.2	run_pairedcars	21
7.1.3	run_intensity_selfcal	22
7.1.4	run_bandpass_selfcal	22
7.1.5	run_pol_selfcal	22

This is the official documentation of the P-AIRCARS. Source code can be found at : <https://github.com/devojyoti96/P-AIRCARS.git>

INTRODUCTION

1.1 P-AIRCARS (Polarimetry using Automated Imaging Routine for Compact Arrays of the Radio Sun)

P-AIRCARS is an automated calibration and imaging routine for polarization calibration and imaging of solar observations done with Murchison Widefield Array (MWA) <https://www.mwatelescope.org>. P-AIRCARS has been developed and is maintained by solar physics group at the National Centre for Radio Astrophysics, Tata Institute of Fundamental Research (NCRA-TIFR), Pune, India <https://www.ncra.tifr.res.in>.

P-AIRCARS uses CASA (Common Astronomy Software Application) <https://casa.nrao.edu> for intensity and band-pass self-calibration and polarisation calibration is performed by our own implementation of the full Jones calibration algorithm described by Mitchell et al. 2008 <https://doi.org/10.1109/JSTSP.2008.2005327>.

Imaging at P-AIRCARS is performed by WSClean <https://wsclean.readthedocs.io>. If WSClean can not be installed, P-AIRCARS performs imaging by CASA. When CASA is used for imaging the computation speed is slow.

Basic philosophy of P-AIRCARS is self-calibration and use the instrumental model to perform the precise calibration. Details of the algorithm and implementation can be found in the following papers

- 1.AIRCARS (Mondal et al. 2019) <https://doi.org/10.3847/1538-4357/ab0a01>
- 2.Kansabanik et al. 2021a, in preparation
- 3.Kansabanik et al. 2021b, in preparation

If you use P-AIRCARS for any publication or scientific work, please cite these three papers.

P-AIRCARS has several modules and scripts. Modules can be used for self-calibration of astronomical observations also. Users who want to use the modules please read the documentation of the module details.

Instructions for installing P-AIRCARS in local machines, laptops and work stations can be found in the installation section. Implementation for high performance computing (HPC) environment is in development and will be available shortly. A basic tutorial is also included for new users.

For any queries and issues reach us at or raise an issue in github:

dkansabanik@ncra.tifr.res.in, paircarsnotification@gmail.com

1.2 System requirements

1.2.1 Minimum hardware requirements

CPU: 8 cores

RAM : 8 GB

Installation space : 700 MB

Disk space : 6 times of the data volume

1.2.2 Workstation configuration

P-AIRCARS has been tested on two types of workstations with following configurations.

1. Intel Xeon Gold 6138 2 GHz 40 CPUs - 256 GB RAM
4 x 10TB (RAID-0) 40TB , 2 sets of 3 x 10 TB (RAID-5) , 2 x 20 TB
2. Boston Intel E5-2699 v3 2.30GHz, 72 CPUs
256 GB (8 x 32) RAM - 2 x 300 GB SSD System
Data 41 TB (12 x 4 TB SATA with RAID-5, 256KB stripe)
Data1 81 TB (12 x 8 TB SATA with RAID-5, 512KB stripe)
960 GFLOPS

1.2.3 Software environment

P-AIRCARS has been tested in the following linux environments

1. CentOS 7
2. Ubuntu 20.0.4

Testing in other linux environments is in progress.

Details of other software requirements is available in installation section.

INSTALLATION

P-AIRCARS is a self-contained python package. It is build on python3. Python version greater than 3.6 is required for P-AIRCARS. It is recommended to use python virtual environment to install P-AIRCARS, but it is not a necessary requirement. Installation steps are as follows.

2.1 Creating virtual environment

Install python virtual environment:

```
>> python3 -m pip install --upgrade pip
```

```
>> python3 -m pip install virtualenv
```

Activate virtual environment:

```
>> python3 -m venv -system-site-packages {path_to_virtualenv}/paircars
```

```
>> source {path_to_virtualenv}/paircars/bin/activate
```

Check the virtual environment path:

```
>> which python3
```

It should show “{path_to_virtualenv}/paircars/bin/python3”

2.2 Obtaining P-AIRCARS

P-AIRCARS source code can be downloaded from <https://github.com/devojyoti96/P-AIRCARS.git>.

P-AIRCARS is not public now. It will be public very soon. If you want to use P-AIRCARS before it becomes public, please reach us at dkansabanik@ncra.tifr.res.in, paircarsnotification@gmail.com

2.3 Installing P-AIRCARS

Move to the P-AIRCARS directory.

Run *setup.py*

```
>> python3 setup.py install
```

That's all. It will automatically install all required packages and libraries. This installation process is in very early stage. If you find any issues during installation, please reach us at dkansabanik@ncra.tifr.res.in, paircarsnotification@gmail.com.

Note: If you installed P-AIRCARS in a virtual environment, please make sure you are inside the virtual environment. P-AIRCARS can not be accessed outside the virtual environment in this case.

>> which python3

It should show “{path_to_virtualenv}/paircars/bin/python3”. If you are not in virtual environment, please activate it.

TUTORIAL

P-AIRCARS is a user-friendly and robust pipeline. In this tutorial we will learn how to run P-AIRCARS on a set of solar observations from MWA in default mode.

3.1 Start the P-AIRCARS

1. Open your terminal
2. Type 'go-paircars' and press enter

```
>> go-paircars
```

3. If user want to use P-AIRCARS to download data from MWA ASVO, follow the steps below:

If user does not have MWA ASVO account :

- a. Go to <https://asvo.mwatelescope.org/>
- b. Click on **REGISTER** button at top-right corner.
- c. Fill up the details and submit.
- d. It may take some days to accept the registration form by MWA ASVO administration.

After the registration is done:

- a. Go to <https://asvo.mwatelescope.org/>
- b. Click on **LOGIN** button at top-right corner.
- c. Give your *username* and *password* and click on *Login*.
- d. After login, click on your name on the top right corner and go to **MY PROFILE**.
- e. In that page user can find the **API key**

Copy the MWA ASVO API key

4. Type 'go-paircars--ASVO_API_key={API_key_of_user}' and press enter.

```
>>go-paircars --ASVO_API_key={API_key_of_user}
```

User only need to supply the MWA ASVO key once and it will be saved in the P-AIRCARS cache directory. If user delete the cache, user needs to supply it again.

3. P-AIRCARS input window will pop up.

The screenshot shows the P-AIRCARS user interface. The background is a photograph of the MWA antenna array. The interface is divided into three main sections: INPUTS, ADVANCED INPUTS, and HPC SETTINGS.

INPUTS

- Data Directory *: Name of the directory of data.....
- Base Directory *: Name of the base directory
- Image Directory: Name of the directory to store final images
- Time range: yy0/mm0/dd0/hh0:mm0:ss0.ff0~yy1/mm1/dd1/hh1:mm1:ss1.ff1
- Channel range: ch0~ch1, ch2~ch3.....
- Caltable Names: /path/to/caltable0,/path/to/caltable1.....
- Robustness: ☐ 0 ☐ 1 ☐ 2 Quality: ☐ 0 ☐ 1 ☐ 2 Verbose: ☐
- Save Logs: ☐ Interactive: ☐ Decorrelation Correction: ☒
- Reference Antenna: 1 Auto-calculate Parameters: ☒
- Use WSClean: ☒ CASA Auto-masking: ☐ Notification: ☒
- CASA Mask: CASA mask path.....
- CASA Mask String: Mask string in CASA format.....
- e-mail: Enter email address to send notifications..... Bandpass: ☒ Polcal: ☒
- Frequency Interval (kHz): 160.0 Time Interval (s): 0.5
- Frequency Width (kHz): 160.0 Temporal Width (s): 0.5
- CPU (%): 50.0 Clear Virtual Screens: ☒
- Save models: ☐ Save residuals: ☐ XY Cutout (degree): 9
- Perform Flagging: ☒ Use aNKFflag: ☒ HPC environment: ☐
- Input file: Load P-AIRCARS input file.....
- Restart P-AIRCARS: ☐ Fresh Start P-AIRCARS: ☒

ADVANCED INPUTS

- Pixelsize: xxarcsec or xxarcmin Number of pixels: Number of pixels
- Multiscale scales: 0,3,9,6 UV-taper: xlambda or xxxlambda
- Start Sigma: 0.0 Sigma Step: 0.5 Minimum Sigma: 0.00
- Residual Flux Fraction: 0.15 UV range: CASA uvrange format
- Skip Frequency (kHz): 1280.0 Skip Time (s): 120.0
- Gain Minimum SNR: 4.0 DR RMS step: DR rms step
- DR Negative step: DR negative step Minimum DR: Minimum DR
- Maximum DR: Maximum DR Minimum Selfcal SNR: Selfcal SNR
- Extra time averaging (s): Extra time Maximum average time (s): Max time
- Weighting mode: uniform/natural/briggs Robust: 1.0

HPC SETTINGS

At the bottom, there are logos for MWA and NCRA-TIFR, and buttons for Validate Inputs, Save Inputs, Run P-AIRCARS, Open Log, and Download Data.

3.2 Download the data

Click *Download Data* on P-AIRCARS user interface.

A popup window will open.

The screenshot shows a popup window titled "Download MWA data". It contains the following fields and buttons:

- MWA ASVO API key *: 85143177-2c22-4ebe-ac3c-0aaa6dfc5e14
- Data Directory *: Name of the directory of data
- Time range *: yy0/mm0/dd0/hh0:mm0:ss0.ff0~yy1/mm1/dd1/hh1:mm1:ss1.ff1
- Download calibration data: ☒
- Project ID: G0002
- Observation ID: OBSID1,OBSID2
-
-

Fill up the details and click on *Start download*. Data jobs will be submitted and waiting for download. A new popup window will show the progress of the data download. Data will be saved in the *Data Directory*.

3.3 Fill up the inputs

3.3.1 Manual input

Two inputs are mandatory

1. *Data directory* : Name of the directory where all measurement sets to be imaged are present. User can type the full directory path in the entry box or use the browse button to select the directory.
2. *Base directory* : Name of the base directory where all the calibration and imaging will be done. User can type the full directory path in the entry box or use the browse button to select the directory.

3.3.2 Load from previous input file

If a previous P-AIRCARS input file with '.paircars' exists, inputs can be filled automatically from that input file. To do that follow the steps

1. Press the *load* button at the right of the **Input file** at the bottom of the INPUTS block and select the input file.
2. Select the file and click *open* and it will automatically load the input file and fill the inputs.

3.4 Validating inputs

After filling the inputs either manually or from a previous input file, user can validate if all inputs are correct or not.

1. Click **Validate Inputs** button at the bottom of the INPUTS.
2. If inputs are correct a pop up message will be shown saying, "Inputs are correct".

3.If mandatory inputs are missing it will show "Mandatory inputs are missing".

Check if all mandatory inputs are given or not.

3.5 Save inputs

If user wants P-AIRCARS inputs can be saved in a '.paircars' file.

1. Click **Save Inputs** button at the bottom.
2. A file dialog box will pop up. Default directory is your present directory and default file name is 'inputs.paircars'. User may change the directory and file name.
3. Click on *save* to the inputs.
4. If the input file saved successfully a pop up message will show the location where the input file saved. Otherwise it will show error.

3.6 Run the P-AIRCARS

After giving all the mandatory inputs, we are now ready to start the P-AIRCARS.

1. Click on **Run P-AIRCARS** button at the bottom.
2. If P-AIRCARS is already running or interrupted at the same base directory, a pop up message will ask ‘P-AIRCARS already running in the base directory. Do you really want to over run?’
3. If you want to re-run P-AIRCARS in the same base directory overriding the present running P-AIRCARS, press *yes* , otherwise press *no*.
4. If you press *yes*, P-AIRCARS will start. It may take some time to start. When started a pop up message will show ‘P-AIRCARS has started. Job ID : your_job_id’.
5. Note down the Job ID. You can use this Job ID to track the progress of the P-AIRCARS.
6. If you have started P-AIRCARS, it will start after pressing **Run P-AIRCARS** button and pop message will show the Job ID.

3.7 Notification over e-mail

User can get the updates about the progress of p-AIRCARS in their e-mail. To enable this feature follow the steps

1. Select the *Notification* button.
2. Put the e-mail id of the user in the *e-mail* entry box.

3.8 Locate the final images

If you give some custom path to save final images in the *Image Directory*, final images will be saved there. Otherwise, final images will be saved in a sub-directory named ‘final_images’ inside the *Base Directory*.

DETAILS OF INPUTS

In this section the details of the P-AIRCARS inputs are given.

4.1 Inputs

1. **Data Directory** : Name of the directory where all measurement sets to be imaged are present. If user want to run in *offline* mode, please be sure that all metafits files corresponding to all measurement sets are also present in the *Data Directory*. This is a mandatory input.
2. **Base Directory** : Name of the directory where all calibration and imaging analysis will be done. Either give an existing directory, otherwise P-AIRCARS will create the *Base Directory* in the given path. This is a mandatory input.
3. **Image Directory** : Name of the directory to save final images. If no input is given final images will be saved in a sub-directory named 'final_images' inside the *Base Directory*.
4. **Time range** : Time ranges to be imaged. Time range format is hh0:mm0:ss0.fff~hh1:mm1:ss1.fff,hh2:mm2:ss2.fff~hh3:mm3:ss3.fff,...
5. **Channel range** : Channel ranges to be imaged. Channel range format is ch0~ch1,ch2~ch3,...
6. **Caltable Names** : Prior caltable names obtained from calibrator observations. Give the full paths separated by commas.
7. **Robustness** : Determine the robustness of the calibration solutions and self-calibration convergence. It takes three values 0, 1, 2 with increasing robustness. Default value is 1.
8. **Quality** : Determine the image quality. It takes three values 0, 1, 2 with increasing quality of the images. Default value is 1.
9. **Verbose** : If selected, verbose output logs will be saved. Also all the images, measurement sets and calibration tables are also saved. In verbose mode amount of disk space requirements will be more. Turn on only for debugging purposes. Default it is turned off.
10. **Save Logs** : If selected, logs for self-calibration and imaging rounds will be saved. Default it is turned off.
11. **Interactive** : By default it is turned off. If selected, user can do interactive cleaning and also can change inputs manually during runtime. Only useful for specific debugging.
12. **Decorrelation Correction** : If selected, perform the amplitude decorrelation correction. By default it is switched on. Details of the correction can be found in Mondal et al. 2020 <https://doi.org/10.3847/1538-4357/ab0a01>
13. **Reference Antenna** : Select the reference antenna from first 30 core antennas. Default is 1.

14. **Auto-calculate Parameters** : If selected, P-AIRCARS will calculate calibration and imaging parameters automatically from the data. By default, it is switched on. When it is turned off, user can give their own calibration and imaging parameters in advanced inputs. Only experienced users should turn this off.
15. **Use WSClean** : Use WSClean for imaging. By default it is switched on. If it is turned off, CASA will be used for imaging. Imaging using CASA is significantly slow compared to WSClean. Thus users are instructed to use WSClean. If WSClean is not installed, P-AIRCARS will automatically use CASA for imaging even if Use WSClean is turned on.
16. **CASA Auto-masking** : When CASA is being used for imaging, CASA auto-masking can be used by enabling this option. By default it is not activated.
17. **Notification** : If selected, P-AIRCARS will automatically send the updates about the progress of calibration and imaging to the user specific e-mail id.
18. **CASA Mask** : When using CASA for imaging, user may supply a CASA mask file here. By default it is not activated. It will be activated when Use WSClean is turned off.
19. **CASA Mask String** : CASA soecific mask string when using CASA for imaging. By default it is not activated. It will be activated when Use WSClean is turned off.
20. **e-mail** : User e-mail address to send notifications. If no e-mail id is given, no notification can be sent.
21. **Bandpass** : Whether perform bandpass self-calibration or not. By default it is turned on.
22. **Polcal** : Whether perform polarisation calibration and imaging. By default it is turned on. When it off, P-AIRCARS will only make Stokes-I images.
23. **Frequency Interval** : Frequency interval to make images. By default it is 160 kHz.
24. **Time Interval** : Time interval to make images. By default it is 0.5 s.
25. **Frequency Width** : Bandwidth to average. By default it is 160 kHz.
26. **Time Width** : Temporal averaging. By default it is 0.5 s.
27. **CPU(%)** : Percentage of total available cpus to be used. By default it is 50%.
28. **Clear Virtual Screens** : Clear the virtual screens opened during the P-AIRCARS run.
29. **Save models** : If selected, save final model images. By default it is turned off.
30. **Save residuals** : If selected, save final residual images. By default it is turned off.
31. **XY cutout** : Size of the final image cutout. Default size is 3 degree.
32. **Perform Flagging** : Perform flagging during the calibration or not. By default it is on.
33. **Use aNKflag** : If selected, use aNKflag for flagging. If aNKflag is not installed, do not use.
34. **HPC environment** : Switch on if running is high performance computing (HPC) environment. When it is on fill the settings for HPC environment.
35. **Input file** : Load the P-AIRCARS inputs from previous saved P-AIRCARS input file. Extension of P-AIRCARS input file is '.paircars'. After loading inputs will be filled automatically.
36. **Restart P-AIRCARS** : Restart P-AIRCARS from the interrupted step. It is off by default.
37. **Fresh start P-AIRCARS** : Start P-AIRCARS in fresh. If selected, all previous analysis in the same *Base Directory* will be deleted. By default it is switched on.

4.2 Advanced Inputs

Advanced input block will be activated when *Auto-calculate Parameters* is switched off. Use *Advanced Inputs* carefully and suggest to use only by experienced users.

1. **Pixelsize** : Pixel size of the image in arcsec or arcmin, e.g. '20.0arcsec' or '1.0arcmin'.
2. **Number of pixels** : Total number of pixels in the image. Product of *Pixelsize* and *Number of pixels* determine the size of the field-of-view.
3. **Multiscale scales** : Multiscale scales in units of number of pixels.
4. **UV-taper** : UV taper string in CASA format
5. **Start Sigma** : Starting value of sigma for rms based thresholding in self-calibration.
6. **Sigma Step** : Stepsize to reduce the sigma value with self-calibration iterations.
7. **Minimum Sigma** : Minimum value of sigma to reduce during self-calibration.
8. **Residual Flux Fraction** : Fraction of flux remained in the residual image to stop the self-calibration.
9. **UV range** : UV range in CASA format for self-calibration.
10. **Skip Frequency** : Frequency step to skip during calibration. Calibration will be performed after *Skip Frequency* steps.
11. **Skip Time** : Time step to skip during calibration. Calibration will be performed after *Skip Time* steps.
12. **Gain Minimum SNR** : Minimum SNR of the gain solutions to be used. Gain solutions less than minimum SNR will be rejected.
13. **DR RMS step** : If rms based dynamic range increases less than this value, self-calibration converged.
14. **DR Negative step** : If negative based dynamic range increases less than this value, self-calibration converged.
15. **Minimum DR** : Minimum dynamic range to save a final image.
16. **Maximum DR** : Maximum dynamic range required. If the imaging dynamic range reached this value, self-calibration stopped.
17. **Minimum Selfcal SNR** : Minimum antenna based SNR required to allow the self-calibration to start.
18. **Extra time averaging** : If large calibration solutions flagged, average over *Extra time* to gain SNR. The value is in second.
19. **Maximum average time** : Maximum allowed value of the *Extra time*.

4.3 High Performance Computing (HPC) settings

ROBUSTNESS AND QUALITY

In this section two major input parameters for P-AIRCARS, *Quality* and *Robustness*, will be discussed.

5.1 Quality (Q)

Quality parameter determines the quality of the final images. With the increase of *Quality* value, antennas will be added in smaller steps, self-calibration will be continued for smaller value of residual fluxes and final CLEANing of the images will be deeper. *Quality* takes three values; 0, 1, 2. Default value is 1.

5.2 Robustness (R)

Robustness parameter determines the robustness of the calibration and imaging. Increasing the robustness parameter will use strong convergence criteria for self-calibration, uses higher value of minimum SNR for gain solutions, performs calibration at smaller time and frequency intervals. *Robustness* takes three values; 0, 1, 2. Default value is 1.

The calibration and imaging parameters are determined based on the combination of *Quality* and *Robustness*. The calibration and imaging parameters for each of the combinations with increasing significance are listed below. With the increasing significance, the calibration and imaging quality will be better, but the computational time will also increase. Default value of 1 for *Quality* and *Robustness* is an optimization between final image quality and computational time.

5.3 Combinations

Q = 0, R = 0

residual_frac = 0.03

start_sigma = 9.0

min_sigma = 6

gain_minsnr = 3.0

DR_delta_rms = 25

DR_delta_neg = 20

min_DR = 20

max_DR = 100

skip_time = 480 s

skip_freq = 2560 kHz

Q = 0, R = 1

residual_frac = 0.03

start_sigma = 9.0

min_sigma = 7

gain_minsnr = 3.0

DR_delta_rms = 22

DR_delta_neg = 18

min_DR = 25

max_DR = 500

skip_time = 240 s

skip_freq = 2560 kHz

Q = 0, R = 2

residual_frac = 0.03

start_sigma = 9.0

min_sigma = 8

gain_minsnr = 3.0

DR_delta_rms = 20

DR_delta_neg = 15

min_DR = 30

max_DR = 1000

skip_time = 120 s

skip_freq = 2560 kHz

Q = 1, R = 0

residual_frac = 0.015

start_sigma = 10.0

min_sigma = 7

gain_minsnr = 4

DR_delta_rms = 20

DR_delta_neg = 15

min_DR = 30

max_DR = 1000

skip_time = 120 s

skip_freq = 1280 kHz

Q = 1, R = 1 (Default)

residual_frac = 0.015

start_sigma = 10.0
min_sigma = 8
gain_minsnr = 4
DR_delta_rms = 18
DR_delta_neg = 12
min_DR = 35
max_DR = 5000
skip_time = 60 s
skip_freq = 1280 kHz

Q = 1, R = 2

residual_frac = 0.015
start_sigma = 10.0
min_sigma = 9
gain_minsnr = 4
DR_delta_rms = 15
DR_delta_neg = 10
min_DR = 40
max_DR = 10000
skip_time = 30 s
skip_freq = 1280 kHz

Q = 2, R = 0

residual_frac = 0.01
start_sigma = 11.0
min_sigma = 8
gain_minsnr = 4.5
DR_delta_rms = 18
DR_delta_neg = 12
min_DR = 40
max_DR = 10000
skip_time = 15 s
skip_freq = 640 kHz

Q = 2, R = 1

residual_frac = 0.01
start_sigma = 11.0
min_sigma = 9
gain_minsnr = 4.5

DR_delta_rms = 15
DR_delta_neg = 10
min_DR = 45
max_DR = 50000
skip_time = 12 s
skip_freq = 640 kHz
Q = 2, R = 2
residual_frac = 0.01
start_sigma = 11.0
min_sigma = 10
gain_minsnr = 4.5
DR_delta_rms = 12
DR_delta_neg = 8
min_DR = 50
max_DR = 100000
skip_time = 10 s
skip_freq = 640 kHz

PAIRCARS

6.1 paircars

6.1.1 paircars.access_ms

6.1.2 paircars.basic_func

6.1.3 paircars.decor

6.1.4 paircars.dynamic_spectrum

6.1.5 paircars.flagger

6.1.6 paircars.fullpol_selfcal_LTS

6.1.7 paircars.intensity_selfcal_LTS

PAIRCARS SCRIPTS

7.1 paircars scripts

7.1.1 control_paircars

Usage: PAIRCARS master controller for each day calibration

Options: -h, --help show this help message and exit

 --basedir=Directory path, Name of base directory for a given day

7.1.2 run_paircars

Usage: Perform self calibration of a single time and frequency slice

Options: -h, --help show this help message and exit

 --msname=Measurement Set, Name of measurement set of a single time and frequency slice

 --metafits=Metafits file, Name of metafits file of the observation

 --basedir=Directory path, Name of the base directory

 --workdir=Directory path, Name of the working directory

 --ref_freq_avg=Float, Frequency averaging for reference ms

 --ref_time_avg=Float, Time averaging for reference ms

 --ref_time_freq=Boolean, Reference measurement set or not

 --do_bandpass=Boolean, Perform bandpass calibration or not

 --do_polcal=Boolean, Perform polarisation calibration or not

 --cal_attenuation=Float, Attenuation in dB for calibrator observation

 --caltables=String, comma separated, Previous calibration tables

 --scratch=Boolean, Start from scratch or not for reference time frequency slice

 --wsclean=Boolean, Use WSClean for imaging or not

 --cpu_frac=Float, Fraction of cpu to use

7.1.3 run_intensity_selfcal

Usage: Perform intensity self calibration of a single time and frequency slice

Options: -h, --help show this help message and exit

- msname=Measurement Set, Name of measurement set of a single time and frequency slice
- metafits=Metafits file, Name of metafits file of the observation
- workdir=Directory path, Name of the working directory
- dopoint=Boolean, Want to try with point source model
- verbose=Boolean, Verbose mode
- fresh=Boolean, Start fresh self calibration loop
- reduce_flags=Boolean, Try to reduce flag solutions if it is more than 5%
- scratch=Boolean, Start from scratch or not for reference time frequency slice
- caltables=String, comma separated, Previous caltables
- wsclean=Boolean, Use WSClean for imaging or not

7.1.4 run_bandpass_selfcal

Usage: Perform bandpass self calibration

Options: -h, --help show this help message and exit

- msname=Measurement Set, Name of measurement set of a single time and frequency slice
- metafits=Metafits file, Name of metafits file of the observation
- workdir=Directory path, Name of the working directory
- verbose=Boolean, Verbose mode
- fresh=Boolean, Start fresh self calibration loop
- caltables=String, comma separated, Previous caltables
- wsclean=Boolean, Use WSClean for imaging or not

7.1.5 run_pol_selfcal

Usage: Perform polarisation self calibration of a single time and frequency slice

Options: -h, --help show this help message and exit

- msname=Measurement Set, Name of measurement set of a single time and frequency slice
- metafits=Metafits file, Name of metafits file
- workdir=Directory path, Name of the working directory
- verbose=Boolean, Verbose mode
- fresh=Boolean, Start fresh self calibration loop
- gaincal=Boolean, Perform gaincal using leakage corrected model (Only do when no calibrator observation is present)
- caltables=String, comma separated, Previous caltables

--wsclean=Boolean, Use WSClean for imaging or not