EFOSC2_Scripts

User Document (v1.0)

Document written by Adam Higgins and Klaas Wiersema

Contents

1 Overview

This document will help explain the scripts found at **https://github.com/abh13/EFOSC2_Scripts**. We will go through the workings of the scripts (with references to specific methods) and output details. All scripts are written in Python3 and specific modules are required to run the scripts (outlined below). Once you have them working feel free to analyse your data with them! We ask that you kindly cite our paper (Wiersema et al. 2018) if you publish any data using the **EFOSC2_Pol_Calc.py**. If anything is unclear or if you have any suggestions/feedback, feel free to email us at abh13@le.ac.uk or k.wiersema@warwick.ac.uk. We hope you find the document and the scripts useful!

2 EFOSC2_Pol_Calc.py

The EFOSC2_Pol_Calc.py script acts as a pipeline for converting flux measurements from reduced EFOSC2 imaging polarimetry frames to calibrated polarisation measurements. EFOSC2, mounted at one of the Naysmyth platforms on the New Technology Telescope (NTT), is based on the EFOSC instrument and is capable of performing imaging and spectroscopy. Using wave plates and a Wollaston element, one is able to perform polarimetry. For details on the instrument capabilities see ESO 2016 Manual. For full details on the experimental set-up and methods outlined in this manual please refer to Wiersema et al. 2018 and references within.

NOTE: this script takes $\approx 2-3$ mins to run on a standard laptop/desktop.

2.1 Required python modules

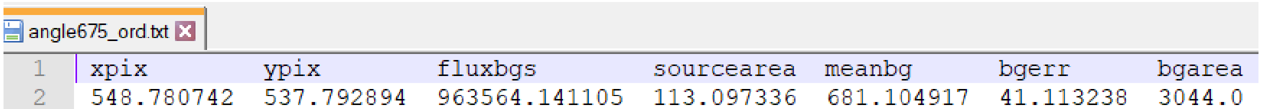
- os
- sys
- argparse
- numpy
- pandas
- scipy
- matplotlib
- emcee
- corner

2.2 Pre-script method

When observing a target source using EFOSC2 an exposure is taken using a set angle for the half-wave plate and a Wollaston prism which splits the light beam into two beams – the ordinary and extraordinary. We use the "Woll_Prism20" prism to allow sufficient space between the two light beams which allows us to extract a larger local background annulus. This is done for each of the 4 following half-wave plate angles: $0°$, $22.5°$, $45°$ and $67.5°$. Four wave plate angles are used instead of two to increase the polarimetric accuracy (see Patat & Romaniello 2006 for details).

EFOSC2 exhibits additional instrumental polarisation away from the optical axis. This script DOES NOT account for that so we recommend placing your source target on a CCD pixel such that this effect is negligible - Wiersema et al. 2018 for our method of choosing a suitable pixel. We use the following naming convention: we name the upper beam image the ordinary beam and the lower beam image the extraordinary beam.

You should perform your own photometry for each of the four half-wave plate angles, resulting in eight output files - the photometry for the ordinary and extraordinary beams at each angle. Please ensure your photometry output files follow the same format as ours (see figure 1) and are saved as .txt ASCII data files - see section 2.3 for more details on the file naming convention.



angle675_ord.txt

	xpix	ypix	fluxbgs	sourcearea	meanbg	bgerr	bgarea
1	xpix	ypix	fluxbgs	sourcearea	meanbg	bgerr	bgarea
2	548.780742	537.792894	963564.141105	113.097336	681.104917	41.113238	3044.0

Figure 1: Typical output file containing the photometric properties of the target source. Columns contain x-pixel CCD coordinate, y-pixel CCD coordinate, total source flux after background subtraction, the area of the source aperture, the mean background flux, the error on the background flux and the area of the rectangular background annulus (from left to right).

2.3 Running the script

To run the script simply type the following command into your terminal - **python EFOSC2_Pol_Calc.py DIR FILTER PAR** [−−**gain**] where **DIR** is the directory containing the eight .txt files, **FILTER** is the filter used for the observations (note: we only covered V, B and R in Wiersema et al. 2018 but the script additionally calibrates for the U and i filters), **PAR** is the parallactic angle in degrees and −−**gain** is an optional argument to manually chose the detector gain (default is set at 1.1 electrons per ADU). For additional help append -h to the command above. The eight text (.txt) files described above should have the following names:

- angle0_exord.txt
- angle0_ord.txt
- angle225_exord.txt
- angle225_ord.txt
- angle45_exord.txt
- angle45_ord.txt
- angle675_exord.txt
- angle675_ord.txt

representing the half-wave plate angle and the respective beam images.

2.4 Polarisation Measurements

To calculate the degree of polarisation for the chosen source within the input data files we follow the methods described in Patat & Romaniello. 2006. The polarisation measurements in this script are represented by the components of Stokes vectors. If you are unsure on how the polarisation values and associated errors are derived we recommend reading that paper along with the source code as they do a fantastic job of describing the method step by step!

2.5 Calibration of Polarisation

The setup of EFOSC2 introduces instrumental polarisation which has to be accounted for and removed from the measured polarisation to find the true value.

Two methods were used to calibrate the instrumental polarisation by fits onto the observed Q, U values from a set of unpolarised and polarised standard stars; a semi-analytical sinusoidal fit (Heidt

& Nilsson. 2011) and a Mueller Matrix approach (Giro et al. 2003). Although we discuss both in the paper, and both can be seen in the source code, the Mueller Matrix approach is more rigorous and is used in the Pipeline. We left the sinusoidal fit code in for a comparison of the methods.

This script utilises MCMC sampling using the package EMCEE (Foreman-Mackey et al. 2013) to estimate the detector offset angle and refractive index multiplication factor of the EFOSC2 M3 mirror which dominates the instrumental polarisation behaviour. Occasionally one of the MCMC walkers can get stuck in a local minimum, producing a poor corner plot - the script usually will give a warning. This happens because the initial parameters have only 20 walkers and 2500 steps whose start positions are randomly distributed throughout the parameter space (this allows for future expansion into more wavebands i.e. where we don't know the offset and multiplication factor). As we are only estimating 2 parameters these numbers are fine most of the time and it keeps the computational time at only a couple of minutes. Gelman-Rubin statistics are displayed to quantitatively determine if the chains converge. If the chains don't converge run the script again! As an extra note, reflective boundaries are introduced at $\pm 90°$. This is required as the offset angle is double degenerate with two angles exactly 180° apart. This isn't a real degeneracy - the angles are the same and both produce the same results. They just represent the same angle in a different angle reference system but the MCMC walkers don't know that! Please refer to Wiersema et al. 2018 for a full description.

You need the following source files to calibrate the instrumental polarisation. **Please ensure these files are in the same directory as the script**:

- pb_standards.txt – contains the names of the stars, the parallactic angles, the measured polarisation values of q, $qerror$, u, $uerror$ for the unpolarised standard stars and the filter wavelength in microns (0.440 - B band).

- pv_standards.txt contains the names of the stars, the parallactic angles, measured polarisation values of q, $qerror$, u, $uerror$ for the unpolarised standard stars and the filter wavelength in microns (0.547 - V band).

- pr_standards.txt – contains the names of the stars, the parallactic angles, the measured polarisation values of q, $qerror$, u, $uerror$ for the unpolarised standard stars and the filter wavelength in microns (0.643 - R band).

- pu_standards.txt – contains the names of the stars, the parallactic angles, the measured polarisation values of q, $qerror$, u, $uerror$ for the unpolarised standard stars and the filter wavelength in microns (0.355 - U band).

- pi_standards.txt – contains the names of the stars, the parallactic angles, the measured polarisation values of q, $qerror$, u, $uerror$ for the unpolarised standard stars and the filter wavelength in microns (0.793 - i band).

- METALS_Aluminium_Rakic.txt – contain complex refractive index values for aluminium at various wavelengths.

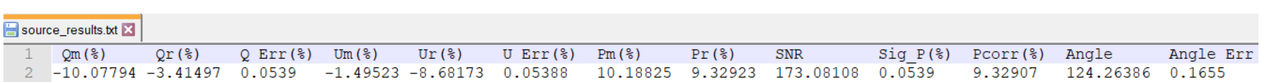
The standard star observations were taken between 5-8 August 2018 (we updated the files from our 2016 run to values taken from our latest run) and are used to estimate the effective refractive index which will differ somewhat from the theoretical aluminium values, because of the effects of oxidisation, dust accumulation, etc. As the mirror surface is continuously changing these measurements are only valid for $\approx$ 3 - 4 weeks around the observed dates (see Covino et al. 2014). Therefore we strongly recommend taking a few unpolarised standard star observations of your own to accurately calibrate both the multiplication factor and detector offset at the time of your observations - using our standard star observations could affect your polarisation measurements by up to 0.5% (see Wiersema et al. 2018 for details on this effect). Create your own versions of the tables following our naming convention. Please ensure that the standard star measured polarisations are in terms of the normalised Stoke's parameters q and u.

These files are in the same GitHub folder as the scripts under **EFOSC2_Pol_Files.zip**. Once the instrumental polarisation is accounted for the script will calculate the correct polarisation for the target source.

2.6 Polarisation Output

The script writes the results of the pipeline to a file called **source_results.txt**. The results file contains many values which are defined below and an example can be seen in figure 2:

- Qm – measured Stoke's q parameter (%) without removing instrumental polarisation.
- Qr – actual measured value of sources q parameter ($\times$100%).
- Qerr – error on Qr (1σ).
- Um – measured Stoke's u parameter ($\times$100%) without removing instrumental polarisation.
- Ur – actual measured value of sources u parameter ($\times$100%).
- Uerr – error on Ur (1σ).
- Pm – measured value of degree of polarisation ($\times$100%) without removing instrumental polarisation.
- Pr – measured value of degree of polarisation ($\times$100%) intrinsic to the source.
- SNR – signal to noise ratio of the source given by Pr/Sig_P.
- Sig_P - the standard error on P (only to be used for confidence intervals when SNR > 3).
- Pcorr – Pr corrected for bias using the MAS estimator described in Plaszczynski et al. (2015).
- Ang - Estimated polarisation angle (ignore the angle and error if the polarisation is an upper limit).
- Ang Err - Estimated standard error on the polarisation angle.



source_results.txt

Qm(%)	Qr(%)	Q Err(%)	Um(%)	Ur(%)	U Err(%)	Pm(%)	Pr(%)	SNR	Sig_P(%)	Pcorr(%)	Angle	Angle Err
-10.07794	-3.41497	0.0539	-1.49523	-8.68173	0.05388	10.18825	9.32923	173.08108	0.0539	9.32907	124.26386	0.1655

Figure 2: An example polarisation results output file. Please see above for a discussion on the output values.

2.7 Additional Output

The script will also output the Matrices for the incoming light from the M2 to the M3 mirror with a 0° reflection and the M3 mirror with a 45° reflection.

The script will automatically save a plot showing the best-fit model for the Q and U Stokes parameters instrumental polarisation against parallactic angle (called best_model.png). It will also save the walker paths and corner plot of the multiplication factor and detector offset to file (called walker_paths.png and pol_param_corner.png respectively).

3 EFOSC2_Image_Red.py

The **EFOSC2_Image_Red.py** script is used to reduce the raw EFOSC2 imaging and polarisation science files. The script produces a master bias and master flat frame and then calibrates the science image using these frames.

3.1 Required python modules

- os
- sys
- numpy
- glob
- astropy
- argparse

3.2 Pre-script method

The script is designed to use the calibration and science fits files available on the ESO archives (found here - http://archive.eso.org/eso/eso_archive_main.html).

Before running the script, you should have the individual flat and bias frames required to produce the master frames in the same directory as the raw science image fits file. The bias frames are the same for both polarisation and imaging with BIAS as the object keyword. The flats for polarisation should be DOME flats and the imaging flats should be SKY,FLATS as given by the OBJECT keywords. The polarimetry flats were taken with a continuously rotating waveplate - see Wiersema et al. 2018 for details.

NOTE: The individual bias, flat and raw science fits files should be of the array shape, 1030x1030 (NAXIS1 and NAXIS2 keywords) and the binning should be the same as for the science data (CDELT1 and CDELT2) keywords. The script is set at 2x2 binning by default but can easily be modified.

3.3 Running the script

The script is designed to run directly from the command line. For a description of the required and optional input variables run the command **python EFOSC2_Image_Red.py -h**.

3.4 Reduction and output

The script can perform three tasks that produce the following files:

- **Master bias frame** - This will be created from the individual bias frames present in the directory and saved to a file name of your choice. The master bias frame is produced by stacking together the individual bias frames and calculating the mean value per pixel through the stack.

- **Master flat frame** - This will be created from the individual dome or sky flat frames present in the directory and saved to a file name of your choice. We first stack the individual flat frames together - for sky flats, sigma clipping is applied to each individual frame to remove any star contamination and the the master bias frame is removed - for dome flats, the master bias is simply removed from each individual frame. The mean value per pixel through the stack is calculated and the pixel means are used to create a single stacked frame. The median of the stacked frame over the entire CCD is calculated and the stacked frame is then divided through by the median to create the master flat.

- **Reduced science fits file** - This will be created by subtracting the master bias frame from the science image and then dividing through by the master flat frames created above (naming convention - FB_XXXX.fits where XXXX.fits is the raw science fits file).

If you have previously produced master bias and flat frames, providing you use their file names in the input the script will use these when creating the reduced science fits file.

NOTE: The output master bias, master flat and reduced science files are of the shape 1024x1024. The pixels numbered 1024-1030 in both axis are clipped off. The justification for this is to remove the very noisy edge pixels that can affect how well the master flat and bias frames are produced.

4 EFOSC2_Pol_Phot.py

The **EFOSC2_Pol_Phot.py** script takes the reduced science images and performs photometry on each one with the results used in the **EFOSC2_Pol_Calc.py**. In Wiersema et al. 2018 we performed the photometry using the IRAF package APPOLA. The photometry included in this script was intended be used for a quick science analysis but gives the same result as, and a similar level of precision as APPOLA for sources with decent SNR. As discussed in section 2 we recommend that you should perform your own photometry and therefore this script is therefore just included for completeness.

4.1 Required python modules

- argparse
- os
- sys
- numpy
- pandas
- scipy
- astropy
- photutils
- matplotlib

4.2 Pre-script method

The automated photometry included in this script was designed to perform photometry on one source placed at the centre of the optical axis (see section 2.2 for discussion). If the field of view is very crowded or the source is very faint, the detection software may not select the required target or detect an uneven number of sources per half-wave plate image. We therefore recommend that you perform your own photometry. You can then calculate the polarisation of your source(s) using the **EFOSC2_Pol_Calc.py** script - please ensure your photometry output files follow the same format as ours (see figure 1).

4.3 Running the script

After the standard data reduction (bias subtraction, flat fielding etc.) you should have a reduced fits file for each of the four wave plate angles. To run the script simply type the following command - **python EFOSC2_Pol_Phot.py DIR** [−−**ap**] [−−**fwhm**] where **DIR** is the directory containing the four fits files, −−**ap** represents an optional argument for choosing an aperture diameter (default set at 2.0×FWHM) and −−**fwhm** represents an optional argument to manually choose the FWHM (default set to auto-estimate). For additional help append -h to the command above. The four fits files described above should have the following names:

- 0ang.fits
- 225ang.fits

- 45ang.fits
- 675ang.fits

4.4 Source Photometry

This script utilises the photutils package to detect the desired source within the input fits files and calculate the source flux and background contribution. The script is set up to only search in a small(ish) area near the center of the CCD/optical axis (as described above) and to detect two sources - the ordinary and extraordinary images of the target source.

The FWHM of the source, if not manually chosen, is estimated by calculating the distance between the pixel containing the peak flux of each source, and the nearest pixel where the flux has decreased to half of the maximum. This is done for both the x and y axes. The FWHM is then used to create an aperture to calculate the source flux. A large rectangular aperture is also produced and placed around the source to measure the local background count rate. The mean local background rate is then combined with the area of the aperture to estimate the background rate contribution within the source aperture. This can be further optimised with small changes to the script.

4.5 Photometry Output

The photometry part of the script produces a number of outputs. Firstly, eight output files are produced containing the photometry properties of the source (the ordinary and extraordinary beams for each half-wave plate angle). They have the following naming convention - **angXXX_ord.txt** and **angXXX_exord.txt** where XXX is the wave-plate angle. This is the same naming convention as section 2.3. Figure 1 shows the format of a typical output file. A further image is produced for each half-wave plate angle with the detected sources and the background area (see figure 3). From here you can run **EFOSC2_Pol_Calc.py** to calculate the polarisation of the sources.

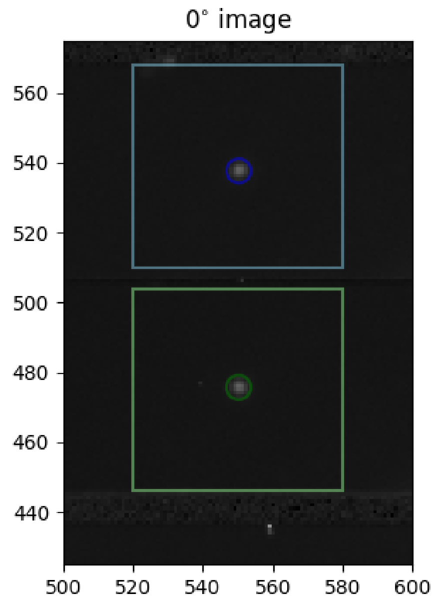


Figure 3: Image output file showing the ordinary and extraordinary beam source images produced by the Wollaston prism. The sources have been detected correctly and the aperture and annulus are of an appropriate size. For reference, the ordinary beam is at the top.

References

Covino, S. et al. 2014, Astrono. Nachr., 335, 117–123

Foreman-Mackey, D., Hogg, D. W., Lang, D., Goodman., J. 2013, PASA 125, 306

Giro, E., Bonoli, C., Leone, F., Molinari, E., Pernechele, C., Zacchei, A. 2003, SPIE 4843, 456

Heidt, J. & Nilsson, K. 2011, A&A 529, A162

Patat, F. & Romaniello, M., 2006, PASP 118, 146

Plaszczynski, S., Montier, L., Levrier, F., Tristram, M., 2015, MNRAS, 439, 4048

Simmons, J. F. L., Stewart, B. G., 1985, A&A 142, 100

Wiersema, K. et al. 2012, MNRAS 426, 2

Wiersema, K. et al. 2014, Nature 509, 201

Wiersema, K., Higgins, A. B., Covino, S., & Starling, R. L. C., 2018, PASA 35, e012