A simulator for POLLUX

Simona Lombardo (LAM)

The POLLUX simulator can be downloaded at https://github.com/Simlomb/PolluxSimTools .

In this package there are several modules, one can for example:

1. Run the web-ETC locally from a terminal as: `bokeh serve --show pollux_etc/`
2. Run the full simulator from a terminal as: `python main_pollux.py`

The web-ETC takes as input a series of parameters (aperture size, redshift, AB-magnitude of the object, exposure time) that can be set by the user for several possible SED templates and provide plots with the signal-to-noise ratio (S/N) for the selected observing channel/mode for POLLUX (NUV, MUV, FUV, both in spectropolarimetric and spectroscopic-only mode).

The chosen detectors for POLLUX are EMCCDs, which when operated in Electron Multiplication mode (EM) provide an additional gain, G, up to $\times 1000$ and an additional noise term, the Excess Noise Factor (equal to $\sqrt{2}$ for gain of 1000). For an input spectrum (list of templates available, at the end of this document) of flux φ (in $counts/s \cdot cm^2 \cdot \text{Å}$), background equivalent flux, BEF (in $counts/s \cdot cm^2 \cdot \text{Å}$), exposure time t_{exp}, the S/N equation used is:

$$\frac{S}{N} = \frac{I}{\sqrt{(I + B + D) \cdot ENF^2 + RON^2 \cdot N_{pix}}}$$

Where $I = \varphi \cdot A_{eff} \cdot t_{exp} \cdot \Delta\lambda$ is the input flux in counts as observed by POLLUX, $B = BEF \cdot A_{eff} \cdot t_{exp} \cdot \Delta\lambda$ is the astrophysical background (Earth-glow and interplanetary Lyα) in counts as observed by POLLUX, $A_{eff} = R_L^4 \cdot A_L \cdot T_{Pollux} \cdot QE_{EMCCD}$ is the effective area of the POLLUX channel considered, $D = D_c \cdot t_{exp} \cdot N_{pix}$ is the total dark noise in counts and $RON = RN/G$ is the readout noise of the EMCCD operated in EM mode.

In the previous equation we considered that the EMCCDs were used in imaging mode, we also considered that $\Delta\lambda = \lambda/R$ where R is the resolution of POLLUX, R_L is the reflectivity of the LUVOIR's mirrors and A_L is its aperture area, T_{Pollux} is the efficiency of the POLLUX channel considered, QE_{EMCCD} is the quantum efficiency of the related EMCCD, D_c and RN are its dark and the readout noise respectively, and finally N_{pix} is the number of detector pixels within the FWHM of the object observed.

If the observations are made with the photon counting mode of the detectors, then the equation is:

$$\frac{S}{N} = \frac{DP \cdot I}{\sqrt{\left(I + B + D + C \cdot N_{pix} \cdot N_{frame}\right) \cdot DP}}$$

Where DP is the detection probability, C is the clock induced charge noise, that becomes dominating when reading out fast, N_{frame} is the number of frames required to have a total exposure time of t_{exp}. For these calculations we assumed a $DP = 0.75$ for a gain of 1000, with a threshold at 250 counts/frame.

The simulator `main_pollux.py` also allows to generate 2D images of POLLUX. These images show, for each channel, the positions of the echelle spectra generated (also for the two polarization states if the observing mode is with polarizers in). The simulator includes the BEF, the blurring effect due to the instrumental Point Spread Function (PSF), and some of the effect of the noise terms, such as gain non uniformity, readout and dark noise, saturation. The images produced are generated for the imaging mode and the photon counting mode of the EMCCDs.

For the simulation of the S/N and the 2D image the user can insert a name with which the plots and the data files generated are saved in a directory within the package (/pollux_tool/). In the 2D image case, also a map of the wavelength position across the EMCCD is saved.

A list of the options available for the simulator is accessed with `main_pollux.py -h`

SED TEMPLATE AVAILABLE AS INPUT:

Input for the simulator	Template used
o5v	O5V Star
hr1886	B1V Star
alplyr	A0V Star
alpcmi	FI5V-V
g2v	G2V Star
gamdra	K5III
mdwarf	M1 Dwarf
mdwarf2	M3 Dwarf
ctts2	Classical TTauri
g191b2b	G191B2B (WD)
gd71	GD71 (WD)
gd153	GD153 (WD)
qso	QSO
s99	10 Myr Starburst
orion	Orion Nebula
nodust	Starburst, No Dust
ebv6	Starburst, E(B-V)=0.6
syfrt1	Seyfert 1
syfrt2	Seyfert 2
liner	Liner
flam	Flat in F_Lambda