

# PlatoSim: an end-to-end PLATO camera simulator for modelling high-precision space-based photometry

N. Janssen<sup>1</sup>, J. De Ridder<sup>1</sup>, D. Seynaeve<sup>1</sup>, S. Regibo<sup>1</sup>, R. Huygen<sup>1</sup>, P. Royer<sup>1</sup>, C. Paproth<sup>2</sup>, D. Grießbach<sup>2</sup>, R. Samadi<sup>3</sup>, D. R. Reese<sup>3</sup>, M. Pertenais<sup>2</sup>, E. Grolleau<sup>3</sup>, R. Heller<sup>4</sup>, S. M. Niemi<sup>5</sup>, J. Cabrera<sup>6</sup>, A. Börner<sup>2</sup>, S. Aigrain<sup>7</sup>, J. McCormac<sup>8</sup>, P. Verhoeve<sup>5</sup>, P. Astier<sup>3</sup>, N. Kutrowski<sup>9</sup>, B. Vandenbussche<sup>1</sup>, A. Tkachenko<sup>1</sup>, and C. Aerts<sup>1,10,11</sup>

<sup>1</sup> Institute for Astronomy, KU Leuven, Celestijnenlaan 200D bus 2401, 3001 Leuven, Belgium  
e-mail: [nicholas.janssen@kuleuven.be](mailto:nicholas.janssen@kuleuven.be)

<sup>2</sup> Institute of Optical Sensor Systems, German Aerospace Center, Rutherfordstraße 2, 12489 Berlin-Adlershof, Germany

<sup>3</sup> LESIA, Observatoire de Paris, Université PSL, Sorbonne Université, Université Paris Cité, CNRS, 5 place Jules Janssen, 92195 Meudon, France

<sup>4</sup> Max-Planck-Institut für Sonnensystemforschung, Justus-von-Liebig-Weg 3, 37077 Göttingen, Germany

<sup>5</sup> European Space Agency/ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands

<sup>6</sup> Institute of Planetary Research, German Aerospace Center, Rutherfordstr. 2, 12489 Berlin, Germany

<sup>7</sup> Sub-department of Astrophysics, Department of Physics, University of Oxford, Oxford OX1 3RH, UK

<sup>8</sup> Department of Physics, University of Warwick, Gibbet Hill Road, Coventry, CV4 7AL, UK

<sup>9</sup> Thales Alenia Space, 5 All. des Gabians, 06150 Cannes, France

<sup>10</sup> Department of Astrophysics, IMAPP, Radboud University Nijmegen, PO Box 9010, 6500 GL Nijmegen, The Netherlands

<sup>11</sup> Max Planck Institute for Astronomy, Koenigstuhl 17, 69117 Heidelberg, Germany

Received 19 April 2023 / Accepted 13 October 2023

## ABSTRACT

**Context.** PLANetary Transits and Oscillations of stars (PLATO) is the ESA M3 space mission dedicated to detect and characterise transiting exoplanets including information from the asteroseismic properties of their stellar hosts. The uninterrupted and high-precision photometry provided by space-borne instruments such as PLATO require long preparatory phases. An exhaustive list of tests are paramount to design a mission that meets the performance requirements and, as such, simulations are an indispensable tool in the mission preparation.

**Aims.** To accommodate PLATO's need of versatile simulations prior to mission launch that at the same time describe innovative yet complex multi-telescope design accurately, in this work we present the end-to-end PLATO simulator specifically developed for that purpose, namely PlatoSim. We show, step-by-step, the algorithms embedded into the software architecture of PlatoSim that allow the user to simulate photometric time series of charge-coupled device (CCD) images and light curves in accordance to the expected observations of PLATO.

**Methods.** In the context of the PLATO payload, a general formalism of modelling, end-to-end, incoming photons from the sky to the final measurement in digital units is discussed. According to the light path through the instrument, we present an overview of the stellar field and sky background, the short- and long-term barycentric pixel displacement of the stellar sources, the cameras and their optics, the modelling of the CCDs and their electronics, and all main random and systematic noise sources.

**Results.** We show the strong predictive power of PlatoSim through its diverse applicability and contribution to numerous working groups within the PLATO mission consortium. This involves the ongoing mechanical integration and alignment, performance studies of the payload, the pipeline development, and assessments of the scientific goals.

**Conclusions.** PlatoSim is a state-of-the-art simulator that is able to produce the expected photometric observations of PLATO to a high level of accuracy. We demonstrate that PlatoSim is a key software tool for the PLATO mission in the preparatory phases until mission launch and prospectively beyond.

**Key words.** methods: numerical – space vehicles: instruments – instrumentation: photometers – planets and satellites: detection

## 1. Introduction

Thanks to the parts-per-million (ppm) precision photometry delivered by space telescopes in the past two decades, the astrophysical frontier has undergone a revolution. As a consequence of this technological progress, answers to profound questions are now within reach, such as the habitability of other planets beyond our Solar System. With the Convection, Rotation, and planetary Transits (CoRoT; [Auvagne et al. 2009](#)) mission marking the start of this endeavour, the quest for habitable planets was followed by NASA's *Kepler* space mission ([Borucki et al. 2010](#)) aimed to discover the first Earth-like planet orbiting a Sun-like star.

Together with *Kepler*'s extended operation, the so-called *K2* mission ([Howell et al. 2014](#)), and successor, the Transiting Exoplanet Survey Satellite (TESS; [Ricker et al. 2015](#)) mission, a wealth of scientific discoveries opened gateways to research areas and synergies thought impossible at the beginning of the millennium. Beyond the continuous planet discoveries by TESS, the ongoing CHARACTERISING EXOPLANET SATELLITE (CHEOPS; [Benz et al. 2021](#)) mission likewise plays a key role in the precise characterisation of known exoplanet systems.

Also complementary science cases have flourished from the long and uninterrupted observations by CoRoT and *Kepler*, and shorter baseline but all-sky coverage observations by TESS.

Being a principle driver for many of these missions, the ability to finally detect acoustic oscillations within the interior of solar-like stars (e.g. Kjeldsen & Bedding 1995) was not only a success story for asteroseismology (e.g. Michel et al. 2008; Gilliland et al. 2010; Chaplin et al. 2010, 2011; Hekker 2013; Huber et al. 2013), but vital for studies of exoplanets (e.g. Christensen-Dalsgaard et al. 2010; Silva Aguirre et al. 2015; Campante et al. 2016), stellar activity (e.g. García et al. 2010; Chaplin & Basu 2014; Brun et al. 2015; Kiefer 2022), and galactic archaeology (e.g. Stokholm et al. 2019; Silva Aguirre et al. 2020; Hon et al. 2021; Stello et al. 2022). The last two decades of space data (initiated with the MOST mission; Walker et al. 2003) have also been a treasure for pulsation studies across the entire Hertzsprung-Russell diagram – from the pre-main sequence to the last stages of stellar evolution, and from the lowest to the highest stellar masses (e.g. see the reviews of Brown & Gilliland 1994; Cunha et al. 2007; Aerts et al. 2010; García & Ballot 2019; Bowman 2020; Aerts 2021).

Despite the remarkable achievements from planet hunting space missions, no Earth-Sun analogue has been discovered to date (e.g. Hill et al. 2023) and the parameter landscape of low-mass and long-orbital period planets is vastly unexplored (e.g. Bryson et al. 2021). Due to the limited sky coverage of CoRoT and *Kepler*, both targeting stars too faint to efficiently follow up with ground-based radial velocity (RV) surveys, the hunt for small terrestrial exoplanets by similar long-baseline missions is soon to be continued with the PLANetary Transits and Oscillation of stars (PLATO; Rauer et al. 2014, and in prep.) mission. PLATO is the third medium (M3) mission in ESA’s Cosmic Vision 2015–2025 programme with a current launch date set for the end of 2026. Compared to its predecessors, PLATO aims to obtain high precision and continuous photometric time series of more than 245 000 bright stars ( $V$ -band magnitude  $< 15$ ) over its nominal mission duration of 4 yr. Using the tools of asteroseismology and ground-based spectroscopy as an integrated part of the mission strategy, the goal of PLATO is not only to detect but also to characterise exoplanets around stars of magnitude  $V < 10$  with a precision of 5% in radius and 10% in mass. Moreover, PLATO is the first mission dedicated to derive the age of planets to a 10% precision from asteroseismic modelling of the host stars. To meet its scientific objectives, PLATO has been designed to provide a photometric precision of  $\leq 50 \text{ ppm h}^{-1/2}$  for more than 15 000 solar-like stars with  $V \leq 11$  (Rauer et al. 2014, and in prep.).

PLATO’s requirements are challenging due to numerous noise sources. These involve complex interactions between the components of the instrument which can best be modelled at pixel level. Thus, prior to the in-flight operations, end-to-end simulations of the instrument have proven to be a very efficient way to scrutinise performance bottlenecks for the success of the mission. Consequently, instrument simulators have been developed for missions covering a wide range of applications, in the X-ray (e.g. SIXTE; Dauser et al. 2019), optical (CHEOPSim; Futyan et al. 2020, SIMCADO; Leschinski et al. 2016), (near)infrared (MIRISim; Klaassen et al. 2021, SpecsIm; Lorente et al. 2006), and all the way to the radio (pyuvsim; Lanman et al. 2019). Also multi-purpose software packages exist, such as MAISIE (O’Brien et al. 2016) and SOPHISM (Blanco Rodríguez et al. 2018), and simulation frameworks, such as Pyxel (Arko et al. 2022), which are specifically designed for detectors.

To confirm that each performance requirement is within scope, several simulation tools have been developed for PLATO.

The PLATO Light Curve Simulator (PSLS<sup>1</sup>; Samadi et al. 2019) and the PLATO Instrument Noise Estimator (PINE; Börner et al. 2022) are pragmatic yet simplified tools. None of these tools alone provide the ability to simulate all of the expected observations of the PLATO space mission, including image time series, meta data, housekeeping data, and light curves. In this work, we present PLatoSim<sup>2</sup>, a dedicated end-to-end PLATO camera simulator with all of these features.

PLatoSim builds on the heritage of the Eddington CCD Data Simulator (Arentoft et al. 2004, De Ridder et al. 2006) that was developed for the decommissioned Eddington (ESA) and MONS (Danish Space Agency) space missions. The original code was later expanded to meet the demands of generalising simulations for space-borne observatories (such as the ASTRIOD Simulator; Marcos-Arenal et al. 2014a). Aiming at realistic applications to the PLATO mission, Zima et al. (2010) adapted the software into a so-called first generation end-to-end PLATO Simulator. That included a change of software language from IDL to C++ to overcome pre-existing performance bottlenecks. Shortly after PLATO’s selection as ESA’s M3 candidate, Marcos-Arenal et al. (2014b) revisited the software to give it a more modern modular software architecture and expanded its use cases for both the PLATO mission and other (future) photometric missions operating in the optical.

With already existing multi-instrument software packages (such as MAISIE and SOPHISM), and the increasing demand for dedicated yet diverse use cases for the PLATO mission, the development of PLatoSim over recent years has somewhat replaced the mission adaptability aspect with an in-depth applicability for the PLATO instrument. This in turn has resulted in huge advancements of the algorithms implemented and allowed the software to stay up to date with changes ranging from the observational strategy at mission level to the description of the smallest hardware components of the payload. Furthermore, a complete Python wrapper around the generic C++ code has made it easy to configure, set up, and run simulations, which has especially proven valuable for the PLATO mission consortium. PLatoSim has so far been used by multiple teams to estimate the impact of technical or programmatic tradeoffs on the final mission performance, including end of life (EOL) ageing effects, preparation of the data-processing pipelines, preparation of the engineering and scientific calibrations, development and real-time testing of the fine guiding sensor algorithms, among others.

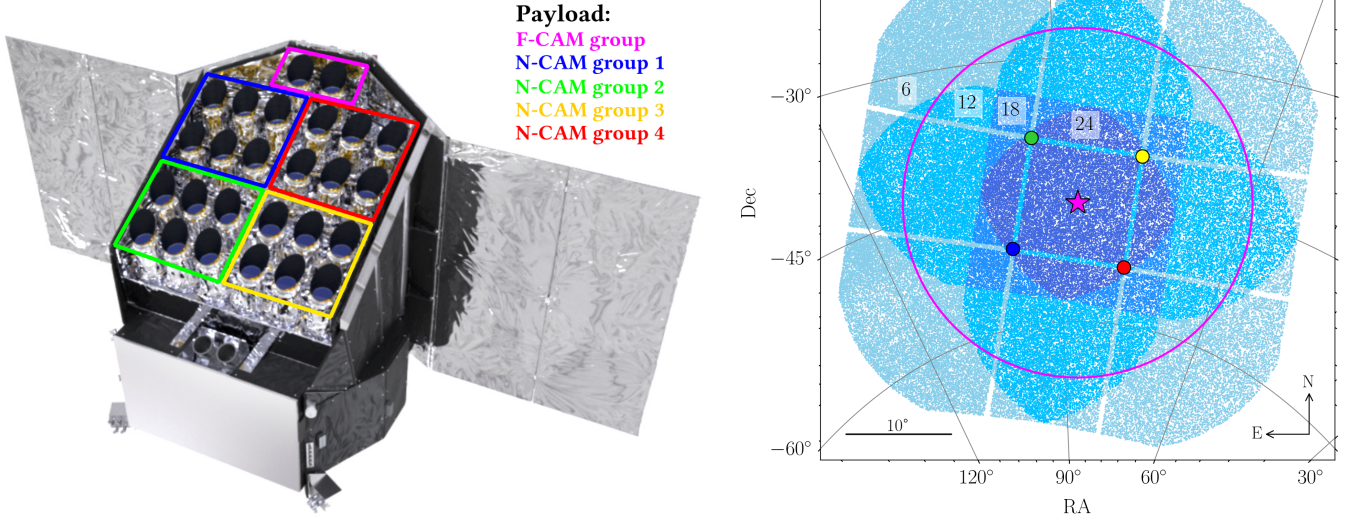
In this paper, we describe the implementation and algorithmic design of PLatoSim, but before doing so a small overview of the payload is given in Sect. 2. Next we present the image acquisition model in Sect. 3, the image generation model in Sect. 4, and each effect implemented will be detailed in Sects. 5 to 8. PLatoSim’s photometry module will be presented in Sect. 9, the software architecture in Sect. 10, applications in Sect. 11, and concluding remarks in Sect. 12.

## 2. The PLATO instrument

As illustrated in the left panel of Fig. 1, the PLATO payload utilises an innovative multi-telescope concept consisting of 26 small but wide-field refractive telescopes ( $\sim 1037 \text{ deg}^2$  each; Pertenais et al. 2021) mounted on a single optical bench.

<sup>1</sup> <https://sites.lesia.obspm.fr/psls/>

<sup>2</sup> <https://github.com/IvS-KULeuven/PlatoSim3/tree/master> (for PLATO mission consortium members).

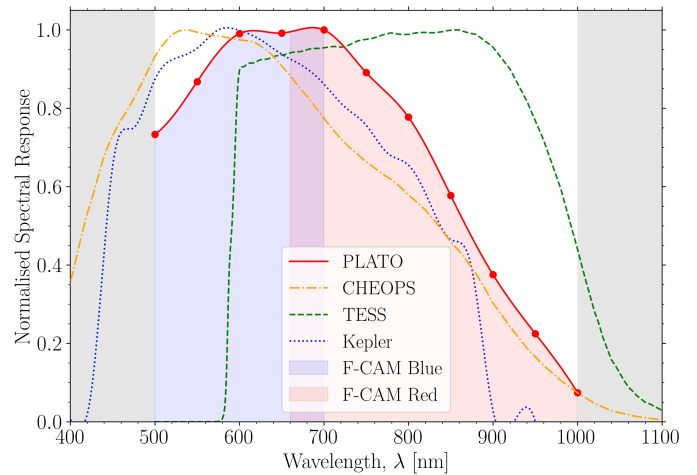


**Fig. 1.** Overview of the PLATO multi-camera design. Left: Schematics of the PLATO spacecraft consisting of the payload module (with colour indication of the telescope groups) and the service module (bus). Credit: ESA/ATG medialab. Right: On-sky FOV of PLATO shown for a pointing towards the Long-duration Observation Phase (LOP) south in equatorial coordinates. The increasing darker shade of blue illustrates the increasing N-CAM overlap of  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$  (also indicated in the white boxes). The coloured dots show the pointing of each N-CAM group, cf. the left-hand plot. The magenta star indicates the pointing of the two F-CAMs, which is parallel to the pointing of the platform, while the magenta circle shows the (camera-only) F-CAM FOV. Data courtesy: [Montalto et al. \(2021\)](#) and [Pertenais et al. \(2021\)](#).

For historical reasons the PLATO mission consortium describes the combined unit of baffle, optical elements, and detectors as a camera and hence for consistency we also adopt this nomenclature.

Each camera consists of a telescope optical unit (TOU) with a 12 cm entrance pupil diameter and a focal plane array (FPA) containing four CCDs connected to electronic controllers called front-end electronics (FEEs) – in total comprising 104 CCDs and 26 FEEs. As visualised in Fig. 1 (left) the cameras are organised in four groups each with six ‘normal’ cameras (or N-CAMs) and one group of two ‘fast’ cameras (or F-CAMs). A preliminary normalised spectral response curve for the N-CAM is shown in Fig. 2, which illustrates a strong similarity in response to the photometer of CHEOPS and *Kepler* as the mission strategy of PLATO targets only slightly cooler (solar) spectral type stars (see e.g. the mission handbooks of CHEOPS<sup>3</sup>, TESS<sup>4</sup>, and *Kepler*<sup>5</sup>)

All cameras of a given group share the same line of sight (LOS) and field of view (FOV). An opening angle of 9.2° of each N-CAM group relative to the F-CAMs, designed to increase the global sky coverage, entails a rather complex overlapping FOV arrangement as shown in the right-hand plot of Fig. 1. This plot shows the provisional long-duration observation phase (LOP) south field which is a subset of the all-sky PLATO Input Catalogue (asPIC; [Montalto et al. 2021](#)). Only the FOV arrangement of the N-CAMs is shown here but the pointing of the F-CAMs is aligned with the pointing of the platform (magenta star). We note that the right-hand plot of Fig. 1 shows the FOV of the spacecraft’s orientation at the first mission quarter. As PLATO will orbit the Sun from the second lagrange point (L2) in exactly 1 yr, the spacecraft is required to realign its solar panels towards the Sun every ~91 days in order to provide power to function. As we



**Fig. 2.** Preliminary normalised N-CAM spectral response curve at beginning of life (BOL; with the red dots representing the mission requirements) compared to those similar planet hunting missions such as CHEOPS (orange dotted-dashed line), TESS (green dashed line) and *Kepler* (blue dotted line). Each response curve is computed with cubic spline interpolation for illustrative purposes. The grey shaded areas are cut-off wavelengths which are dominated by optical transmission in the blue (left) and by the CCD anti-reflection coating in the red (right). The blue and red shaded regions illustrate the blue and red transmission regions of the two F-CAMs, respectively. Data courtesy: ESA and NASA.

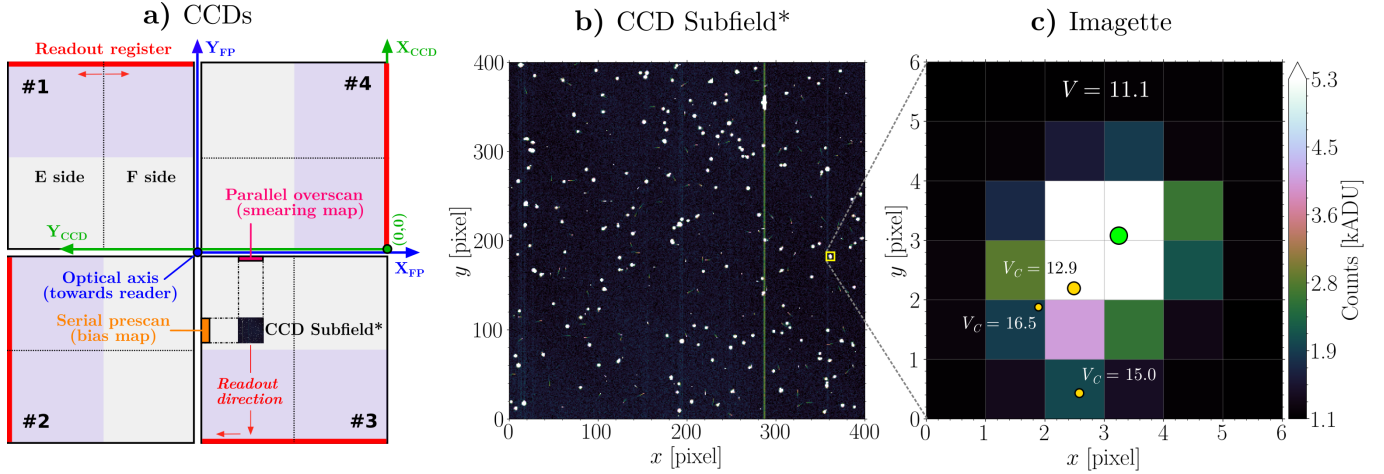
subsequently show in Sect. 11, this is an important constraint for generating realistic simulations.

Depending on the exact location in the FOV a star may be observed with a number of overlapping N-CAMs being  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$  as indicated by the increasing colour gradient from light blue to dark blue. Considering only the effective FOV (i.e. the corrected optical FOV, taking into account the optical and mechanical vignetting and the gaps between the CCDs

<sup>3</sup> <https://sci.esa.int/web/cosmic-vision/-/53541-cheops-definition-study-report-red-book>

<sup>4</sup> <https://heasarc.gsfc.nasa.gov/docs/tess/documentation.html>

<sup>5</sup> <https://archive.stsci.edu/missions-and-data/kepler/documents>



**Fig. 3.** Schematic overview of how PLatoSim generates a CCD image in the FPA. (a) Illustrative overview of the N-CAM FPA with the 4 CCDs. The blue axes indicate the focal plane with the central blue dot (in the middle of the 4 CCDs) represents the optical axis  $Z_{FP}$  pointing in the positive direction towards the reader. The green axes illustrate the origin for  $n_{CCD} = 4$  as a reference and the readout register of each CCD is highlighted with a red bar. Each CCD is divided into an F and E side (with independent CCD and FEE gain appliances) with anti-parallel serial readout directions. For the F-CAM the location of the metallic shields of frame-transfer CCDs are highlighted as purple shaded rectangles. (b) Simulated  $400 \times 400$  pixel subfield of the LOP south. The subfield location on  $n_{CCD} = 3$  is displayed in panel a together with the corresponding parallel overscan and serial prescan region which is used by PLatoSim to reconstruct a proper smearing and bias map, respectively. (c) A so-called imagette showing a zoom-in on a target star from panel b. Indicated by the subpixel barycentres, the PIC target (green dot) has three significant fainter stellar contaminants (yellow dots scaled in size to the target star magnitude).

in the focal plane), the total estimated effective N-CAM FOV of  $2132 \text{ deg}^2$  covers almost 19 times that of *Kepler*.

Sharing identical optical designs the main difference between the F-CAM and N-CAM is the operational mode, the readout cadence, and the wavelength transmission. With a readout cadence of 2.5 s secured by a CCD frame-transfer mode, the F-CAMs are foremost fine guidance sensors for the attitude and orbit control system (AOCS). Featuring frame-transfer CCDs implies that their FOV is about half the FOV of a single N-CAM (Pertenais et al. 2021). Being equipped with respectively a blue and red colour filter makes the F-CAMs ideal science instruments for asteroseismology of bright stars. The N-CAMs operate without an optical filter in a full-frame readout mode at a cadence of 25 s and are the primary photometers used to meet the core science goals.

PLatoSim is designed to model the Teledyne-e2v CCD270 detector that assemble the FPA of PLATO as shown in Fig. 3a. A main characteristic of this detector is the large photosensitive area (see Table 1) together with a division into two CCD halves (a F and E side; see dashed lines) each with an independent readout register to speed up the readout process. On top of this design, the frame-transfer CCDs of the F-CAM are divided into two CCD halves parallel to the readout register: a photosensitive area and a charge storage area. The charge storage area is covered by a metallic shield which is illustrated by the purple shaded regions in Fig. 3a. With the exception of an initial CCD frame-transfer for the F-CAM compared to the N-CAM, each CCD half is read-out following a standard rolling shutter technique first in parallel direction towards the readout register and then in serial direction towards the corner of the register (i.e. left for the F side and right for the E side as seen in the CCD reference frame).

We elaborate more on the details of FPA in the following, however, before doing so, an overview of the PLATO data flow (from acquisition to downlink) is here placed in context

**Table 1.** Characteristics of the PLATO payload.

Parameter	Description/value
TOU	26 units
Optics Design	6 refractive lenses Axis-symmetric dioptics
Aperture diameter	12 cm
Full FOV	$2132 \text{ deg}^2$
Spectral range	500–1000 nm
CCDs (FEEs)	104 units (26 units)
Model Design	Teledyne-e2v CCD270 Back illuminated
Pixel size	$18 \mu\text{m}$
Plate scale	15 arcsec (on-axis)
N-CAMs	24 units
Camera overlap	6, 12, 18, 24 (partial)
Effective FOV	$1037 \text{ deg}^2$
Detector FPA	4 full-frame CCDs
Detector size	$4510 \times 4510$ pixel
Exposure time	21 s
Readout time	4 s
Cadence	25 s
F-CAMs	2 units (red and blue filter)
Camera overlap	2 (full)
Effective FOV	$619 \text{ deg}^2$
Detector FPA	4 frame-transfer CCDs
Detector size	$4510 \times 2255$ pixel
Exposure time	2.3 s
Readout time	0.2 s
Cadence	2.5 s

**Table 2.** PLATO data products downlinked to ground for pre-selected stars at different sampling rates.

Data product	Sampling [s]
Imagette for N-CAM (F-CAM)	25 (2.5)
Light curve: stellar aperture	50, 600
Light curve: inverse aperture	50, 600
Flux centroid: stellar aperture	50, 600
Flux centroid: inverse aperture	50, 600
Flux standard deviation	600
Calibration product	600
Calibration windows	600

**Notes.** Some simple processing is applied to the pixel data before the extraction of for example light curves and flux centroids. We note that only imagettes are downlinked for the F-CAMs.

to PlatoSim. While images are collected by the CCDs, the FEEs are responsible for extracting windows around preselected targets, sky background regions, and CCD regions used for calibration. By analogy with the FEE windowing strategy, the schematic overview of Fig. 3 highlights an important feature which is a part of PlatoSim’s image acquisition model, namely the concept of a CCD subfield. Being a CCD area under consideration of size ( $n_{\text{row}} \times n_{\text{col}}$ ), the subfield is introduced to make long-duration simulation studies of modest sized subfields (such as Fig. 3b) feasible. It is however often more computationally efficient to only simulate an imagette which is a  $6^2$  pixel subfield (or  $9^2$  pixel subfield for the F-CAMs) around a target star (illustrated in Fig. 3c). The imagette is a key data product of PLATO as all targets will have their photometry extracted from an imagette (see Sect. 9). Unlike the designated strategy to only extract imagettes in flight (except for saturated stars), the PlatoSim subfield can take any rectangular shape (smaller than the CCD dimensions) thus allowing more versatile simulations.

Following the data flow post the FEE, all data are sent to the data processing unit (DPU) that extracts and prepares each product for compression, archiving, and lastly transmission to ground. The PLATO data products that will be downlinked to ground are shown in Table 2 and can be categorised into time series of: imagettes, fluxes, flux standard deviations, flux centroids, and data for calibration. To observe as many stars as possible, only imagettes will be sampled at the rate of the nominal N-CAM cadence, while the remaining data products will be sampled every 50 s or 600 s (i.e. averages over two or 24 exposures, respectively). Accompanying these measurements, time series of the inverted aperture mask (called an extended aperture) will be computed as well. The calibration data consist of imagettes (or windows), for example, to compute the local sky background flux, axillary data, and housekeeping data. Except for the flux centroids, all data products of Table 2 can be simulated by PlatoSim as will be addressed in this paper. A high level instrument summary of this section is provided in Table 1.

### 3. Image acquisition model

Every PLATO measurement starts with reading out the CCD subfield after an exposure to obtain the pixel values  $S_{ij}$  expressed in analogue-to-digital units (ADU) where we use  $i$  and  $j$  as the row and column pixel coordinates. These pixel values can be

modelled with

$$S_{ij}(t) = \left[ \left( I_{ij}(t) g_{\text{CCD},ij}(t) + B_{ij}(t) \right) g_{\text{FEE},ij}(t) + \epsilon_{\text{RN},ij}(t) \right]_{n_{\text{bit}}}. \quad (1)$$

Here,  $I_{ij}(t)$  is the number of photo- and thermal electrons accumulated in pixel  $(i, j)$  during the exposure, which is caused by the target star, the sky background, open shutter smearing, dark signal, among others, and which we describe in Sect. 4. The product of the CCD gain  $g_{\text{CCD}}$  (expressed in  $\mu\text{V}/e^-$ ) and the FEE gain  $g_{\text{FEE}}$  (expressed in  $\text{ADU}/\mu\text{V}$ ) make up the total gain  $g$  (expressed in  $\text{ADU}/e^-$ ),

$$g(t) = g_{\text{CCD}}(t) g_{\text{FEE}}(t). \quad (2)$$

We highlight the pixel dependence of the gain: since each CCD are read out in two halves, the left-hand side and the right-hand side have in practice slightly different gains.

Moreover, these gains are not constant but depend on the number of electrons in the well, leading to the well-known effect of CCD non-linearity. The underlying reason is that the CCD output amplifier and the FEE re-amplifier do no longer amplify linearly for a high number of electrons, leading to a sublinear increase of the output voltage (and thus ADU) with an increasing number of electrons. Measurements using the PLATO flight model CCDs revealed that the actual (bias subtracted) pixel signal deviates no more than 100 ADU (<1%) from what a linear model would predict from the number of electrons in the well. This deviation can be reasonably well modelled using a simple polynomial, leading to

$$g_{\text{FEE},ij} = a_0 + a_1 I_{ij} + a_2 I_{ij}^2 + a_3 I_{ij}^3. \quad (3)$$

The model of Eq. (3) accounts both for the CCD non-linearity occurring at low and high flux levels.

The CCD gain and FEE gain also have a weak temperature dependence (of the order of  $10^{-4}$  in units of  $\text{ADU}/\mu\text{V}/\text{K}$  and  $\mu\text{V}/e^-/\text{K}$ , respectively), and may therefore cause a small drift, which is relevant for high-precision photometry. In addition, the PLATO CCDs operate at a temperature of  $-70^\circ\text{C}$  while the FEEs operate around  $30^\circ\text{C}$  and their temperature variations will not be synchronised in time. PlatoSim allows to take this time-dependence into account which can help to define the temperature stability requirements related to the gains.

The quantity  $B_{ij}$  in Eq. (1) is the bias level (expressed in  $\mu\text{V}$ ), that is, the electronic offset produced by the FEE to ensure that the analogue-to-digital (A/D) converter always receives a positive signal. This bias level can be pixel dependent, as is the case for the PLATO detector. Laboratory measurements indicated that the bias is higher at the very first column – the so-called line start effect – and then decreases and levels out to the median value in the subsequent columns at a rate of about 1–2 ADU per 100 pixels. The data reduction pipeline therefore needs to use a more elaborate method to determine the bias than simply using the median pixel value of the prescan region. Simulations that take into account the proper bias pixel dependency are very useful to test this method. In addition, the bias level can be temperature and thus time-dependent. In the case of PlatoSim we allow for a small temperature instability at the level of  $1 \text{ ADU pixel}^{-1} \text{ K}^{-1}$  and the actual bias map for each CCD half is generated for a serial prescan and a (virtual) overscan region of  $25 \times 15$  pixels, respectively (with the former shown in Fig. 3a).

The readout noise  $\epsilon_{\text{RN},ij}$  in Eq. (1) is assumed to be Gaussian,

$$\epsilon_{\text{RN},ij} \sim \mathcal{N}(\mu = 0, \sigma^2 = \sigma_{\text{RN,CCD}}^2 + \sigma_{\text{RN,FEE}}^2), \quad (4)$$

where  $\sigma_{\text{RN,CCD}} = 24.5 \text{ e}^- \text{ pixel}^{-1}$  and  $\sigma_{\text{RN,FEE}} = 32.8 \text{ e}^- \text{ ADU}^{-1}$  are the readout noise levels of the CCD and the FEE at mission BOL, respectively.

The quantity  $[x]_{n_{\text{bit}}}$  in Eq. (1) denotes the quantisation done by the A/D converter, i.e. flooring the quantity  $x$  to the nearest integer value in the range  $[0, 2^{n_{\text{bit}}} - 1]$ . In case of PLATO's  $n_{\text{bit}} = 16$  bit CCDs, this leads to a theoretical upper pixel value of 65 535 ADU. In practice the total gain (expected to be  $g \approx 0.05 \text{ ADU/e}^-$ ) controls the digital saturation, hence saturation is expected to happen around 55 000 ADU and non-linear effects for even lower digital counts cf. Eq. (3).

#### 4. Image generation model

CCD (sub)images are generated with a measurement cadence of 25 s (i.e.  $\Delta t_{\text{exp}} \approx 21 \text{ s}$  exposure and  $\Delta t_{\text{ro}} \approx 4 \text{ s}$  readout) for the N-CAMs, and with a cadence of 2.5 s (i.e.  $\Delta t_{\text{exp}} \approx 2.3 \text{ s}$  exposure and  $\Delta t_{\text{ro}} \approx 0.2 \text{ s}$  frame transfer) for the F-CAMs. Our simulations take into account that the readout of the four detectors in each N-CAM FPA (being four CCDs connected to a single FEE) is synchronised. I.e. all CCDs with the same identification number  $n_{\text{CCD}} \in \{1, 2, 3, 4\}$  across all cameras are read out simultaneously. Depending on  $n_{\text{CCD}}$ , the timing of each CCD readout is delayed by an amount

$$\Delta t_{\text{CCD}} = 6.25 \text{ s} \cdot (n_{\text{CCD}} - 1), \quad (5)$$

due to the spacecraft's limited budget of electrical power, the limited bandwidth between the FEEs and the DPU, as well as to minimise cross-talk between CCDs and FEEs. The same does not hold for the CCDs in the F-CAMs which are read out simultaneously to secure a stable fine guidance and an exact timing of the colour photometry. We note that the time stamps mentioned here are on-board time stamps, not barycentric ones.

The number of electrons  $I_{ij}(t)$  accumulated in a pixel  $(i, j)$  during one observation as explained in Eq. (1) can be divided in two parts: those gathered during the exposure of the CCD and those accumulated during the CCD readout phase as there is no mechanical shutter in a PLATO camera that prevents light from hitting the detector during readout. The latter leads to so-called open shutter smearing and will be explained in more detail in Sect. 5.5. Mathematically we can write

$$\begin{aligned} I_{ij}(t) &= \int_t^{t+\Delta t_{\text{exp}}} I_{ij}^{(\text{exp})}(t') dt' + \int_{t+\Delta t_{\text{exp}}}^{t+\Delta t_{\text{exp}}+\Delta t_{\text{ro}}} I_{ij}^{(\text{ro})}(t') dt' \\ &\approx \sum_n I_{ij}^{(\text{exp})}(t + n \delta t) \delta t + \Delta t_{\text{ro}} \cdot \bar{I}_{ij}^{(\text{ro})}(t), \end{aligned} \quad (6)$$

where  $I_{ij}^{(\text{exp})}$  and  $I_{ij}^{(\text{ro})}$  are the electron accumulation rates during the exposure and readout phase, respectively. The integration over time during the exposure is discretised in time steps  $\delta t$  which we take one-tenth of the time scale of the most rapidly varying time-dependent phenomenon that is affecting the electron accumulation rate (which is usually the spacecraft jitter). Since the readout time  $\Delta t_{\text{ro}}$  is considerable shorter than the exposure time our simulations do not track any time-dependence during the readout phase, but instead we compute an averaged electron accumulation rate and simply multiply it with the readout time, as outlined in more detail in Sect. 5.5.

The number of electrons per second  $I_{ij}^{(\text{exp})}$  accumulated during a CCD exposure can be broken down further into photo-electrons coming from stars  $F_{\star}$ , the sky background  $F_{\text{sky}}$ , stray light  $F_{\text{stray}}$ , together with electrons induced by cosmic particles  $F_{\text{cosmics}}$ , and thermal electrons that are generated even in the absence of incident light, causing the so-called dark current  $F_{\text{dark}}$ . We note that each of these quantities are stochastic, for example  $F_{\star}$  is modelled as a Poisson distributed random variable to include photon noise. Mathematically,  $I^{(\text{exp})}$  can be written as

$$\begin{aligned} I_{ij}^{(\text{exp})}(t) &= \iiint_{i j \lambda} [F_{\star}(t, x, y, \lambda) + F_{\text{sky}}(x, y, \lambda) + F_{\text{stray}}(t, x, y, \lambda)] \\ &\quad \cdot T(t, x, y, \lambda) \cdot E(t, x, y, \lambda) \cdot Q(t, x, y, \lambda) d\lambda dx dy \\ &\quad + F_{\text{cosmic}}(t, i, j) + F_{\text{dark}}(t, i, j), \end{aligned} \quad (7)$$

where the integration happens in two spatial dimensions in the focal plane covering pixel  $(i, j)$  as well as over the wavelength  $\lambda$  in the relevant spectral range. In this expression,  $T(t, x, y, \lambda)$  denotes the optical throughput of the PLATO camera,  $E(t, x, y, \lambda)$  is the detector efficiency (including e.g. the pixel response non-uniformities and bad pixels), and  $Q(t, x, y, \lambda)$  is the quantum efficiency of the CCD. Each of the quantities mentioned above will be described in more detail in this or subsequent sections.

In addition to the electron accumulation described in Eq. (7), there are also several electron redistribution effects that are not as easily described by the expression in Eq. (7). The most relevant ones are charge diffusion in the silicate, charge-transfer inefficiency during readout, and full-well saturation causing blooming. These effects cause a redistribution of electrons in surrounding pixels, and are modelled in PLATOsim as an additional process after the CCD exposure. We refer to Sect. 8 for more details.

Equations (6) and (7) highlight the computational challenge that simulating space-based images poses. PLATO's primary science goal, exoplanets, requires knowledge about instrumental noise in the low-frequency regime (i.e. a time scale similar to both the transit duration and the orbital period of a planet) as well as the noise in the higher-frequency regime (i.e. a time scale similar to several phenomenae of stellar variability). It is therefore sometimes needed to simulate an observational run of 90 days (after which the observations are interrupted to turn the solar panels back towards the Sun). With a measurement cadence of 25 s for the N-CAMs this implies more than 311 000 measurements for a time series, while keeping track of the low-frequency drifts as well as the high-frequency jitter of the spacecraft using a sufficiently small time step  $\delta t$ . The computational burden of the spatial integration over each pixel is driven by the resolution needed to simulate the intra-pixel sensitivity variations, which requires to discretise each pixel in  $64^2$  sub-pixel elements (cf. Zima et al. 2010). The integration over the wavelength range is needed to take into account the variation of the point spread function (PSF) as well as the optical throughput with wavelength.

In practice some simplifications are needed to make the computations feasible. A first approximation is to eliminate the integration over the wavelengths by using a wavelength-averaged PSF weighted with a stellar spectrum of a Sun-like star, as well as using wavelength-averaged values of the throughput  $T$ ,

the detector efficiency  $E$ , and the quantum efficiency  $Q$ . This reduces Eq. (7) to the simplified expression

$$I_{ij}^{(\text{exp})}(t) = \iint_{i,j} \left[ \bar{F}_\star(t, x, y) + \bar{F}_{\text{sky}}(x, y) + \bar{F}_{\text{stray}}(t, x, y) \right] \cdot \bar{T}(t, x, y) \cdot \bar{E}(t, x, y) \cdot \bar{Q}(t, x, y) dx dy + F_{\text{cosmics}}(t, i, j) + F_{\text{dark}}(t, i, j). \quad (8)$$

A second approximation is to neglect the intra-pixel variations and only take into account the pixel variations for those use cases where it has a limited effect.

In the following sections, we provide more details on how we take into account the different quantities included in Eq. (8). Section 5 deals with the incident radiation fluxes  $\bar{F}_\star(t, x, y)$ ,  $\bar{F}_{\text{sky}}(x, y)$ , and  $F_{\text{cosmics}}(t, i, j)$ . This also involves computing the time-dependent focal plane coordinates of the stars, which is described in Sect. 6. Section 7 deals with the throughput and efficiency quantities  $\bar{T}(t, x, y)$ ,  $\bar{E}(t, x, y)$ , and  $\bar{Q}(t, x, y)$ .

## 5. Light and electron sources

This section focuses on part of the ingredients of the image generation described in Sect. 4, more particularly on the polychromatic sources  $\bar{F}_\star(t, x, y)$ ,  $\bar{F}_{\text{sky}}(x, y)$ ,  $\bar{F}_{\text{stray}}(t, x, y)$ ,  $F_{\text{cosmics}}(t, i, j)$ ,  $F_{\text{dark}}(t, i, j)$  in Eq. (8), representing respectively flux originating from incident stellar light, sky background light, stray light, and photoelectrons coming from cosmic hits and dark current.

### 5.1. Incident stellar light

The number of monochromatic photons per second  $F_\star(t, x, y, \lambda)$  in Eq. (7) coming from incident light can be further broken down as

$$F_\star(t, x, y, \lambda) = A \cdot f_\star(\lambda, t) \cdot g(x, y, x_0, y_0, \lambda, t), \quad (9)$$

where  $A$  is the light collecting area of one camera (113.1 cm<sup>2</sup> in the case of a PLATO camera),  $f_\star(\lambda, t)$  is the spectral photon distribution of the star (expressed in photons s<sup>-1</sup> m<sup>-2</sup> nm<sup>-1</sup>) and  $g(x, y, x_0, y_0, \lambda, t)$  is the monochromatic normalised PSF at focal plane coordinates  $(x, y)$  of a star centred around the focal plane coordinates  $(x_0, y_0)$ .

The main time dependence of the PSF comes from a temperature dependence, which can lead to a slight change of the focus. For PLATO, focus changes are dominated by thermal variations of the TOU structure (changing the distance between the lenses), the optical lenses (changing the diffractive index), and temperature differences between the bipods that interface the FPA to the optical bench (Borsa et al. 2022). PlatoSim uses a grid of monochromatic (Huygens) PSFs computed with Zemax OpticStudio<sup>6</sup> with a spatial sampling of 8<sup>2</sup> pixels times 64<sup>2</sup> subpixels per pixel. In-flight, the TOU will be temperature controlled around the pupil of the camera to the optimal focus temperature, and hence from the above discussion, a fixed and homogeneous temperature of -70°C throughout the camera is assumed in the Zemax model. We note that this model realistically reflects the expected optical performance as it includes image distortion together with typical manufacturing and integration tolerances (e.g. refractive index, irregularities, lens and lens surface tilt and/or decentre, and inter-lens distances).

<sup>6</sup> <https://www.zemax.com/pages/opticstudio>

In practice the point spread function of a star is also affected by so-called charge diffusion (see e.g. Rodney & Tonry 2006; Fairfield et al. 2007; Widenhorn et al. 2010; Lawrence et al. 2011). When a photon enters the CCD silicate it frees one or more electrons, which then wander away, including laterally, for a short distance before they are collected by a gate electrode. The net result is that electrons can also end up in neighbouring pixels which slightly diffuses (blurs) the PSF. PlatoSim models this effect by convolving the PSF with a spherical Gaussian having a half-width of 0.2 pixels. In the remainder of this section, when we refer to the PSF we always designate the PSF that has been convolved with a diffusion kernel. Figure 4 illustrates some PSF examples that are used by PlatoSim, both the Zemax as well as the analytical model, with and without charge diffusion taken into account.

As mentioned in the discussion leading to Eq. (8), the integration over the wavelength is computationally cumbersome and in practice PlatoSim therefore uses a polychromatic normalised PSF  $\bar{g}(x, y, x_0, y_0)$  derived as a weighted average of monochromatic PSFs, weighted with the spectral energy distribution (SED) of a G0 dwarf star in the wavelength range of the PLATO passband  $\mathcal{P}$

$$\bar{g}(x, y, x_0, y_0) = \frac{\int_{\mathcal{P}} g(x, y, x_0, y_0, \lambda) f_{\text{GOV}}(\lambda) d\lambda}{\int_{\mathcal{P}} f_{\text{GOV}}(\lambda) d\lambda}, \quad (10)$$

where the SED  $f_{\text{GOV}}(\lambda)$  was taken from Coelho et al. (2005). This allows to approximate Eq. (9) with the polychromatic photon flux

$$\bar{F}_\star(t, x, y) = A \cdot F_\star(t) \cdot \bar{g}(x, y, x_0, y_0), \quad (11)$$

used in Eq. (8). In the expression above  $F_\star(t)$  denotes the polychromatic stellar photon flux integrated over  $\mathcal{P}$

$$F_\star(t) = \int_{\mathcal{P}} f_\star(\lambda, t) d\lambda \approx \Delta\lambda_{\mathcal{P}} \cdot F_0 \cdot 100^{mv(t)/5}. \quad (12)$$

Here,  $\Delta\lambda_{\mathcal{P}}$  is the full width at half maximum of the PLATO passband (about 532 nm for a normal camera),  $V$  is the Johnson-Cousin visual magnitude of the star, and  $F_0 = 1.00179 \cdot 10^8$  photons s<sup>-1</sup> m<sup>-2</sup> nm<sup>-1</sup> is the zero-point reference flux corresponding to a  $V = 0$  G0-dwarf star. Alternatively, it is possible to use PLATO magnitudes  $\mathcal{P}$ , which can be derived from the Johnson-Cousin  $V$  magnitude using the transformation derived from synthetic stellar spectra by Marchiori et al. (2019)

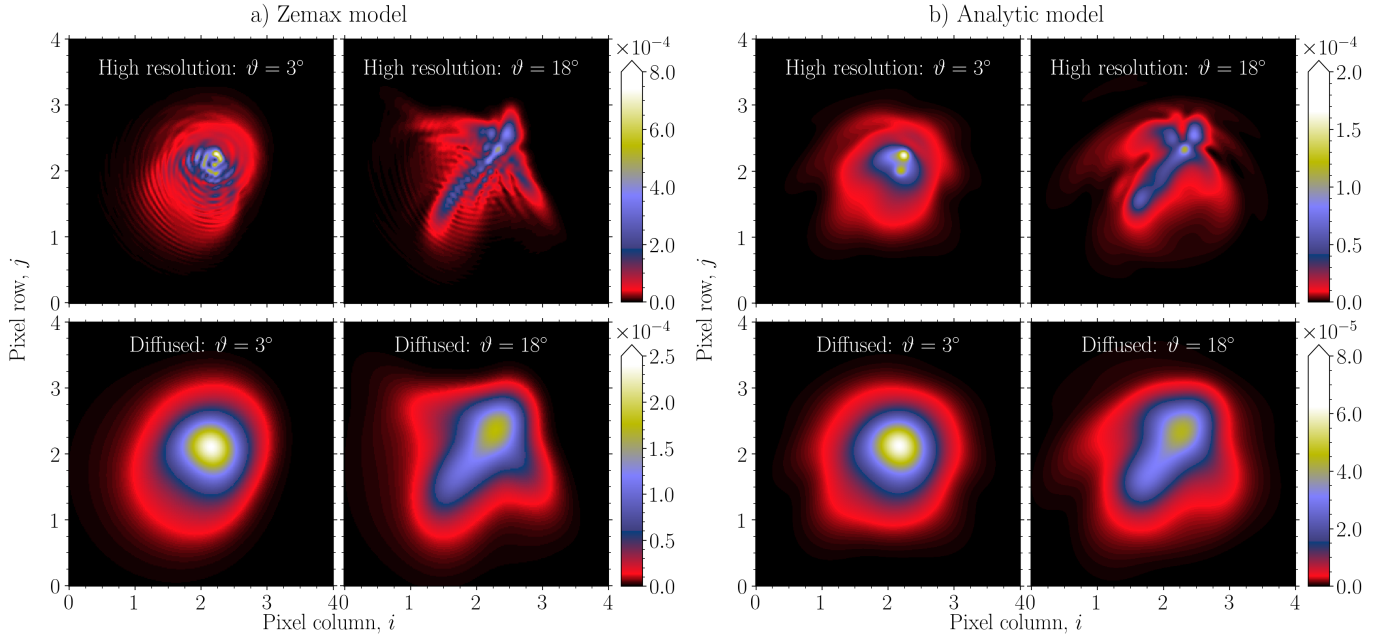
$$V - \mathcal{P} = c_0 + c_1 T_{\text{eff}} + c_2 T_{\text{eff}}^2 + c_3 T_{\text{eff}}^3, \quad (13)$$

where  $T_{\text{eff}}$  is the effective temperature of the star. The best fit coefficients  $\{c_0, c_1, c_2, c_3\}$  tabulated in Marchiori et al. (2019) has recently been revisited by Fialho et al. (in prep.). The flux can then be derived (as was done in the same article) using

$$F_{\mathcal{P}} = 100^{-(\mathcal{P} - \mathcal{P}_{\text{zp}})/5}, \quad (14)$$

with a mission BOL zero point of  $\mathcal{P}_{\text{zp}} = 20.77$  for an A0 dwarf star of  $\mathcal{P} = 0$  being the current best fit estimate for the N-CAM (and correspondingly F-CAM zero-points of  $\mathcal{P}_{\text{zp,blue}} = 20.18$  and  $\mathcal{P}_{\text{zp,red}} = 19.81$  for the blue and red filter, respectively).

The practical implementation of Eqs. (8) and (11) requires the focal plane coordinates  $(x_0, y_0)$  of the star around which the PSF is centred, as well as a numerical integration of the PSF

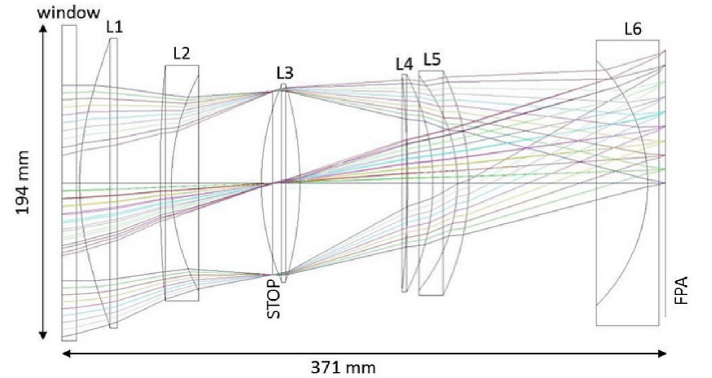


**Fig. 4.** Illustration of a synthetic PLATO PSF generated at different optical axis distances  $\vartheta$  with (a) Zemax OpticStudio and (b) an analytic model. The top panels show the high resolution PSF for  $\vartheta = 3^\circ$  (left) and  $\vartheta = 18^\circ$  (right). The lower panels show the corresponding PSF after a 0.2 pixel Gaussian diffusion kernel has been applied. Each PSF is constructed at an azimuth angle of  $45^\circ$  and has a resolution of 64 subpixel elements corresponding to a  $1' \times 1'$  field on the sky. The image is normalised such that the sum over all pixels is equal to 1.

over the relevant pixels. The former are derived from the equatorial sky coordinates  $(\alpha, \delta)$  of the star using the transformations outlined in Appendix A. However, an accurate derivation also requires to take into account optical distortion, kinematic aberration, the (imperfect) attitude control of the spacecraft, and the thermal drift of the camera, all of which displace the PSF in the focal plane. More details of this description are given in Sect. 6. The integration of the PSF is computationally non-trivial to implement because the PSF varies over the focal plane. Fast convolution of the PSF using the fast Fourier transform (FFT) is therefore only justified when the simulated region of the CCD is sufficiently small. Alternatively, Appendix B shows how an analytical approximation can be constructed that allows to efficiently integrate over the pixels, and which is used by PlatoSim in case of larger CCD regions. Such approximation is also beneficial to realistically characterise the PSF's change in shape and size over time induced typically by a change in the thermal environment, also known as PSF breathing (e.g. see Bély et al. 1993, for orbital focus variations of HST).

On top of all of this, PlatoSim takes into account that the same star is projected multiple times on the focal plane at different locations. Apart from the nominal PSF that carries more than 99.9% of the light, there are two so-called ghost ( $\hat{\rho}$ ) images of the star. The latter are caused by the fact that the optical elements of each camera (as shown in Fig. 5) not only refract the light but also cause internal reflections so that part of the light is also projected elsewhere in the focal plane.

The most important ghost is a point-like image that carries no more than 0.08% of the light. Results from the test campaign of the engineering camera model revealed that the intensity of the point-like ghost decreases exponentially from the optical axis outwards. This ghost image will be extremely weak for stars for which the nominal PSF beyond  $8^\circ$  away from the optical axis, however, in practise heavily saturated stars ( $V \sim 0$ ) will show visible ghosts for nominal PSF positions  $< 12^\circ$  from the optical



**Fig. 5.** Layout of the Telescope Optical Unit (TOU) together with the detectors on the right forming the Focal Plane Array (FPA). Light passes the entrance window to the left (which for the F-CAMs is a dedicated optical filter) and propagates through the refractive optical lenses (L1–L6) unto the FPA on the right. Credit: ESA.

axis (at the level of a few tens of ADU). The point-like ghost is caused after reflection of the light on the CCD surface and on both surfaces of the entrance window (front and back). Its PSF is thus very similar to the nominal one, and is located diametrically opposite of the optical axis, that is

$$\begin{pmatrix} x_{FP} \\ y_{FP} \end{pmatrix}_{\hat{\rho}P} = - \begin{pmatrix} x_{FP} \\ y_{FP} \end{pmatrix}, \quad (15)$$

where we used the optical axis as the origin of the focal plane reference frame. Point-like ghosts are therefore created by stars whose nominal PSF is on another CCD (e.g. see Fig. 3a).

The second ghost image is a so-called extended ghost, and is caused by reflection of the CCD surface and the back of lens L6 (see again Fig. 5). It carries only 0.003% of the light (well below



the mission requirements), and is located on the same CCD as the nominal PSF but radially shifted towards the edge of the FOV

$$\begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \end{pmatrix}_{\text{QE}} = 1.0672 \begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \end{pmatrix}. \quad (16)$$

An analysis with Zemax shows that its PSF can be well approximated with a homogeneous disk with a large diameter (hence the name extended) that depends on the distance of the optical axis, ranging from about 200 pixels close to the optical axis to more than 370 pixels at the edge of the FOV. The exact dependence was tabulated using Zemax, and then approximated with a second order polynomial interpolant that is used in PlatoSim simulations. The nominal source PSF will be inside the extended ghost for a star up to  $6^\circ$  away from the optical axis. As an example, a (saturated) star of  $V = 0$  located  $7^\circ$  from the optical axis will create a ghost of  $\sim 220$  pixel diameter distributed with  $\sim 800 \text{ e}^- \text{ pixel}^{-1}$  (as shown later). Clearly, with such a spatial dilution of a tiny fraction of the light, extended ghosts are only relevant for the brightest stars such as Canopus in LOP south and Vega in the LOP north (following the updated results of Nascimbeni et al. 2022), and will be well below the background noise for all other stars.

### 5.2. Incident light from the sky background

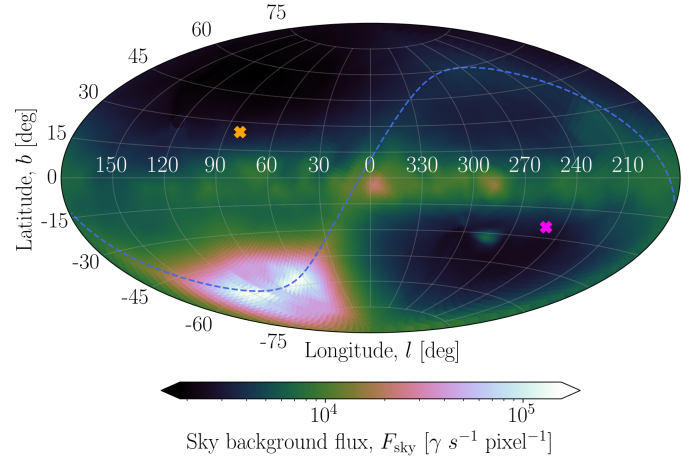
The diffuse sky background  $\bar{F}_{\text{sky}}$  that affects every PLATO CCD measurement consists mainly of zodiacal light, and light from unresolved Milky Way stars. The former is caused by sunlight being scattered by inter-planetary dust particles agglomerated across the ecliptic plane. To model the zodiacal light for simulating space-based photometry, De Ridder et al. (2006) used the monochromatic values of the zodiacal light at  $\lambda = 500 \text{ nm}$  in the vicinity of the Earth tabulated by Leinert et al. (1998), and assumed that the spectral distribution of the zodiacal light is the same as the one of the Sun ( $F_{\odot}(\lambda)$ , tabulated in Wehrli 1985) to estimate the amount of zodiacal light flux that hits the detector.

Marchiori et al. (2019) adopted the same approach but improved upon it by including the reddening factor  $f_{\text{red}}(\lambda)$  of the solar spectrum, the small correction factor  $f_{\text{L2}} = 0.975$  for a spacecraft in L2 rather than the direct vicinity of the Earth, and by including the passband (i.e. the spectral response  $S(\lambda)$ ) of a PLATO camera when integrating over the zodiacal spectrum. PlatoSim adopts the same approach which leads to the following expression for the zodiacal flux

$$F_{\text{ZL}}(\alpha, \delta) = \frac{F_{\text{ZL}}(\alpha, \delta, 500 \text{ nm}) \cdot f_{\text{L2}}}{F_{\odot}(500 \text{ nm})} \int F_{\odot}(\lambda) f_{\text{red}}(\lambda) S(\lambda) d\lambda, \quad (17)$$

where  $F_{\text{ZL}}(\alpha, \delta, 500 \text{ nm})$  is the monochromatic zodiacal flux at 500 nm derived from Leinert et al. (1998). To model the Galactic sky background PlatoSim adopts the same approach as in De Ridder et al. (2006), using tabulated Pioneer 10 observations from beyond 2.8 AU (where the contribution of the zodiacal light is negligible).

Figure 6 shows the combined sky background model of PlatoSim in an all-sky aitoff projection in Galactic coordinates together with the suggested LOPs of Nascimbeni et al. (2022). Since PlatoSim does not include the sky background of extra galactic sources, care must be taken for simulations that covers the FOV of the large Magellanic Clouds (which are partially overlapping with the LOP south). The final selected LOP(s) of



**Fig. 6.** Aitoff projection in Galactic coordinates ( $l$ ,  $b$ ) of the all-sky background model used by PlatoSim. The model includes zodiacal light and diffuse galactic light in units of incoming photons per second per pixel. The blue dashed line shows the ecliptic plane (with the location of the Sun clearly visible as the highest intensity area) and with the crosses illustrating respectively the LOP north (orange) and LOP south (magenta) from Nascimbeni et al. (2022). We note that data gaps in the zodiacal map of Leinert et al. (1998) and in the galactic Pioneer 10 map have been interpolated using a cubic spline.

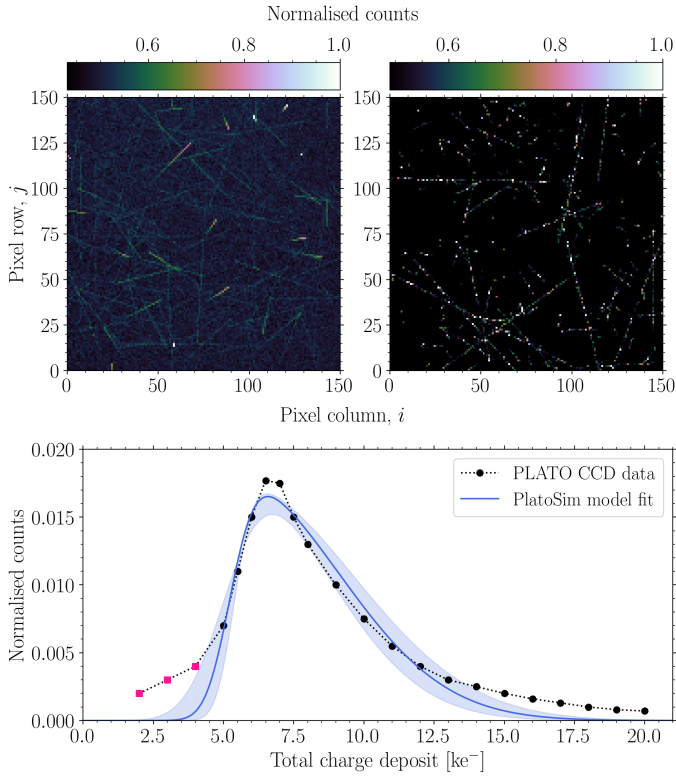
PLATO will be chosen such that the combined sky background flux will be below the mission requirement of  $20 \text{ e}^- \text{ pixel}^{-1} \text{ s}^{-1}$ .

Although the PLATO camera design has a baffle and a stray light mask, these do not perfectly block reflected light of celestial bodies in our Solar System. The main stray light contributors for PLATO are reflected light of the Earth and the Moon, for which the requirement of the combined flux is set to be less than  $40 \text{ e}^- \text{ pixel}^{-1} \text{ s}^{-1}$ . The straylight differs from one camera group to the other, as different camera groups are pointing in a different direction. Detailed modelling of the straylight requires the exact sky position of the Earth and the Moon as well as an optical straylight model resulting from an in-depth analysis of the camera surfaces, coatings, and paintings. Due to the importance of stray light, the inclusion of such a model is currently under development for PlatoSim.

### 5.3. Cosmic rays

Cosmic rays (CRs) are high-energy cosmic particles that leave a high intensity trail over multiple pixels when colliding with a detector. The exact morphology of the dissipated energy on the detector depends both on the properties of detector (e.g. material and front/back illumination) and on the properties of the cosmic particle (e.g. particle type, energy, and incident angle). Furthermore, since the frequencies of cosmic rays depend on the time-varying space weather (dependent on the solar cycle, coronal mass-ejection events, galactic processes, etc.), and their impact strongly depends on spacecraft properties (such as physical orientation, material shielding/penetration, etc.), a realistic model is non-trivial. CR simulators do exist, such as STARDUST (Rolland et al. 2008), Geant4 (Allison et al. 2016), GRAS (Santin et al. 2005), and CosmiX<sup>7</sup> (Lucsanyi & Prod'homme 2020). All of these codes model particle transport using the Monte Carlo technique. CosmiX is the fastest code due to its semianalytical approach. CosmiX was initially developed for space-borne

<sup>7</sup> <https://gitlab.com/david.lucsanyi/cosmix>



**Fig. 7.** CR model comparison. Top panels: A visual comparison between PlatoSim (left) and CosmiX (right), shown for a small 150<sup>2</sup> pixel subfield. The images are generated using a cycle time of 25 s, a CR hit rate of 100 events s<sup>-1</sup> cm<sup>-2</sup> (corresponding to solar maximum), and a CR trail length for PlatoSim of up to 300 pixel. As input for CosmiX, we used a unidirectional 55 MeV proton beam and a 40 μm Si pixel volume. Bottom panel: Number distribution of total charge deposit per event. The black dots show the measurements from the proton irradiation test campaign on a PLATO flight model CCD at 203 K (Prod’homme et al. 2018). The blue solid line is a best fit model of PlatoSim’s skew-normal distribution to the test data and the blue shaded region is the 2σ confidence interval. The first three data points (pink squares) were excluded in the fit as they originate from secondary δ-electrons from the setup. The best fit parameters ( $\psi_{\text{CR}}, \omega_{\text{CR}}, \alpha_{\text{CR}}$ ) = (5232 ± 76, 3842 ± 197, 6 ± 1) were used to produce the PlatoSim simulation above.

missions such as Gaia and PLATO, but is presently too time consuming to be efficiently integrated into PlatoSim.

Instead a simplified model is implemented in PlatoSim: the number of CR hits during an exposure is drawn from a Poisson distribution around a mean value that scales with the exposure time  $t_{\text{exp}}$  and subfield size ( $n_{\text{row}} \times n_{\text{col}}$ ) of the CCD area under consideration

$$N_{\text{CR}} \sim \mathcal{P}(\mu = R_{\text{CR}} \Delta t_{\text{exp}} n_{\text{row}} n_{\text{col}} s_{\text{pix}}^2), \quad (18)$$

where  $R_{\text{CR}}$  is the cosmic hit rate (events s<sup>-1</sup> cm<sup>-2</sup>),  $\Delta t_{\text{exp}}$  is the exposure time, and  $s_{\text{pix}}^2$  is the surface area of one pixel. The impact locations on the CCD are randomly chosen over the entire subfield. The trail length on the CCD of each cosmic hit is drawn from a uniform distribution, and the intensity in electron counts is drawn from a skew-normal distribution (given a location  $\psi_{\text{CR}}$ , scale  $\omega_{\text{CR}}$ , and shape parameter  $\alpha_{\text{CR}}$ ).

We have validated PlatoSim’s CR model to the CosmiX simulator, since this software is open source and in excellent agreement with more complex CR codes such as Geant4 and

GRAS. Since CosmiX is a dedicated module in the open source detector framework Pyxel<sup>8</sup> (Arko et al. 2022), we used this to generate representative PLATO CCD images by loading in a dark frame generated by PlatoSim and then applied CosmiX. A configuration file made it easy to select settings representative of the PLATO CCD needed for CosmiX.

Figure 7 shows a visual model comparison between PlatoSim (top left) and CosmiX (top right). The bottom panel shows a proton irradiation test performed at cold temperature on a PLATO flight model CCD (black dots from Prod’homme et al. 2018) and a best model fit of the PlatoSim skew-normal distribution to the test data (blue solid line). The corresponding CosmiX model (see Lucsanyi & Prod’homme 2020) shows a 93% model agreement to the test data, where the remaining 7% mainly corresponds to so-called secondary δ-electrons emerging from the setup itself (i.e. when the proton beam outside the cryostat collides with the aluminium flange). These electrons leave their imprint as an increased fraction of events below 4 ke<sup>-</sup>, hence, we choose to exclude the first three data points (pink squares) from the PlatoSim model fit.

It is clear that the simplified CR model of PlatoSim naturally shows discrepancies with CosmiX and the PLATO CCD test data (especially at higher deposits of charge). Most noticeable from the pixel data is the difference in CR morphology. In particular CosmiX shows a more discrete nature of the charge deposits along the tracks. The underlying reason is because CosmiX assumes that, while it tracks groups of electron clusters around vertices created by interactions with the incoming ionising particle, the loss of energy of the primary particle through the Si depletion zone of the detector is negligible. On the other hand, PlatoSim models the total charge deposit more continuous and with a gradient that is effectively determined by the distributions from which the energy, incident angle/location, and trail length are drawn. Despite these discrepancies, overall the two models agree sufficiently well in order for PlatoSim to fulfill its original purpose, namely to train the on-board CR rejection algorithm of the PLATO reduction pipeline.

#### 5.4. The dark signal

General for all CCDs, the PLATO detectors show a dark signal, that is thermal electrons that are generated even in the absence of incident light. This dark signal contributes to the total noise budget with a temporal and spatial component which we model. The dark signal accumulated during an exposure of duration  $\Delta t_{\text{exp}}$  and a readout of duration  $\Delta t_{\text{ro}}$ , as occurring in Eqs. (7) and (8), is modelled in PlatoSim using a Poisson distribution

$$F_{\text{dark}}(i, j) \sim \mathcal{P}(\mu = n_{\text{DS},ij} \cdot (\Delta t_{\text{exp}} + \Delta t_{\text{ro}})), \quad (19)$$

where  $n_{\text{DS}}$  is the dark signal. In practice the latter is not fixed for a particular CCD, but shows a fixed-pattern spatial variation over the CCD which is usually characterised by the dark signal non-uniformity (DSNU;  $\sigma_{\text{DSNU}}$ ). PlatoSim models the DSNU by drawing  $n_{\text{DS}}$  from a normal distribution

$$n_{\text{DS}} \sim \mathcal{N}(\mu = \bar{n}_{\text{DS}}, \sigma = \sigma_{\text{DSNU}}). \quad (20)$$

The nominal values of  $\bar{n}_{\text{DS}}$  and  $\sigma_{\text{DSNU}}$  for a PLATO CCD, as tabulated by the manufacturer e2v, are respectively  $\bar{n}_{\text{DS}} = 1.2 \text{ e}^- \text{ s}^{-1}$  and  $\sigma_{\text{DSNU}}/\bar{n}_{\text{DSNU}} = 15\%$  root-mean-square (rms) at mission BOL. These values are slightly conservative compared

<sup>8</sup> <https://esa.gitlab.io/pyxel/>

to on-ground calibration estimates (Verhoeve et al. 2016) of  $\bar{n}_{\text{DS}} \approx 1.03 \text{ e}^- \text{ s}^{-1}$  and  $\sigma_{\text{DSNU}}/\bar{n}_{\text{DSNU}} \approx 12\%$  rms. Apart from the exposure time, the dark signal also depends on the detector temperature, which PlatoSim models as a linear function of the CCD temperature using a slope of  $5 \text{ e}^- \text{ s}^{-1} \text{ K}^{-1}$ , being the mission requirement value.

### 5.5. Readout smearing

Similar to many space-borne instruments the PLATO cameras do not use a mechanical shutter to block light from reaching the CCD during readout. This implies that during the readout the CCD continues to gather photons. During the row transfers at readout, pixels will therefore be shifting ‘under’ the PSFs of stars in the same column and accumulate photons of these stars during the short time that it takes to shift one row of pixels to the next one. In practice this will lead to a uniform bright vertical trail that is imprinted in each column, but is most visible for those columns that contain bright stars. This effect, called readout smearing, is described with the second term of Eq. (6), and is illustrated in Fig. 3b for pixel column  $\sim 290$ , mainly caused by the saturated star in the top of the column.

If, with a slight abuse of notation, we denote with  $I_{ij}^{(\text{exp})}$  the number of electrons that were accumulated in pixel  $(i, j)$  during an exposure of duration  $\Delta t_{\text{exp}}$ , the total number of electrons per second collected in the entire column  $j$  is

$$\bar{I}_j = \frac{1}{\Delta t_{\text{exp}}} \sum_{i=1}^{n_{\text{row}}} I_{ij}^{(\text{exp})}. \quad (21)$$

Here  $n_{\text{row}}$  is the number of rows that are illuminated. We note again that this is different for the F-CAMs than for the N-CAMs as the former feature readout using frame-transfer and the bottom half of their CCDs are covered to prevent illumination. During readout, each pixel  $(i, j)$  in column  $j$  will collect  $\delta t_{\text{trans}} \cdot \bar{I}_j$  electrons, where  $\delta t_{\text{trans}}$  is the time it takes to transfer one row of photoelectrons to the next one. This happens partly during the readout phase of the previous exposure when the row was transferred from the top of the CCD to location  $i$ , and during the readout phase of the current exposure when the row is transferred from location  $i$  to the readout register at the bottom of the CCD.

Since also photons collected during readout are subjected to photon noise, the final value for the accumulated flux during readout is taken from a Poisson distribution

$$I_{ij}^{(\text{ros})} = \mathcal{P}(\mu = \delta t_{\text{trans}} \bar{I}_j). \quad (22)$$

The readout smearing is measured using a parallel overscan region whose size spans 30 virtual pixel rows for the N-CAMs (see pink box of Fig. 3a).

## 6. Focal plane positions of the stars

This section describes how PlatoSim computes the positions  $(x_0, y_0)$  of a star in the focal plane mentioned in Eqs. (9) and (11) and the corresponding pixel positions  $(i_0, j_0)$  on the CCD. A realistic model of the time dependence of these positions is crucial to understand the non-white noise it induces.

### 6.1. Differential kinematic aberration

Given the true equatorial sky coordinates  $(\alpha, \delta)$  of a star, the apparent sky coordinates are slightly different because of stellar

kinematic aberration: the overall motion of the spacecraft relative to the star induces a shift of the apparent stellar position due to the aberration of the light rays as they enter the camera. The amplitude and direction of the shift of the apparent stellar position depends on the angle between the position vector  $s$  of the star and the velocity vector  $v$  of the spacecraft with respect to the star. More specifically, if  $\theta$  is the unaberrated angle between these two vectors (i.e.  $\cos \theta = v \cdot s$ ), then the aberrated angle  $\theta_{\text{ab}}$  between them is given by

$$\theta_{\text{ab}} = \tan^{-1} \left( \frac{\sqrt{1 - \beta^2} \sin \theta}{\beta + \cos \theta} \right), \quad (23)$$

where  $\beta = v/c$  with  $v$  the velocity of the spacecraft with respect to the star and  $c$  the speed of light. The corresponding aberrated position vector of the star  $s_{\text{ab}}$  is given by

$$s_{\text{ab}} = v \cos \theta_{\text{ab}} + \frac{s - v \cos \theta}{|s - v \cos \theta|} \sin \theta_{\text{ab}}. \quad (24)$$

The corresponding pixel displacement of a star therefore depends on its position in the focal plane and varies in time as PLATO’s velocity vector changes over its orbit. However, the AOCS of PLATO uses bright fine guidance stars in the same FOV observed by the F-CAMs to continuously stabilise its pointing. These fine guidance stars experience roughly the same aberration, so that the aberration is therefore largely and continuously corrected by the AOCS. The correction is only approximate since a given pixel’s LOS depends on its exact location in the focal plane and the small pixel-to-pixel variation of the aberration that still occurs. This differential aberration is not radially symmetric around the optical axis of a camera but is offsets from the pointing axis of the platform determined by the F-CAMs. Hence the maximum amplitude of the differential aberration of an N-CAM is at the FOV edge furthest from the platform pointing, resulting in a shift of up to  $\sim 0.8$  pixel in 3 months.

The differential aberration is taken into account using a realistic orbit read by PlatoSim. Generally PLATO orbits around the L2 following a libration point orbit (a so-called Lissajous orbit), but the dominant velocity component for the kinematic aberration is the one following the orbit of L2 around the Sun. In the remainder of the paper, when we use sky coordinates, we always refer to the apparent sky coordinates subjected to kinematic aberration.

### 6.2. Projection of the star on the focal plane

The exact position of the star on a CCD depends on where in the sky the spacecraft platform is pointing, how the camera is mounted on the platform, how the focal plane reference frame is defined inside the camera, and finally how the CCDs are orientated inside the focal plane. In practice, PlatoSim models the CCD coordinates  $(x_{\text{CCD}}, y_{\text{CCD}})$  through a set of reference frame transformations

$$\begin{pmatrix} x_{\text{CCD}} \\ y_{\text{CCD}} \\ 1 \end{pmatrix} = \mathbf{R}_{\text{FP}}^{\text{CCD}} \cdot \mathbf{R}_{\text{CAM}}^{\text{FP}} \cdot \mathbf{R}_{\text{PLM}}^{\text{CAM}} \cdot \mathbf{R}_{\text{EQ}}^{\text{PLM}} \cdot \begin{pmatrix} x_{\text{eq}} \\ y_{\text{eq}} \\ z_{\text{eq}} \end{pmatrix}. \quad (25)$$

Here,  $(x_{\text{eq}}, y_{\text{eq}}, z_{\text{eq}}) = (\cos \delta \cos \alpha, \cos \delta \sin \alpha, \sin \delta)$  are the components of the unit vector pointing towards a star with apparent sky coordinates  $(\alpha, \delta)$  in the equatorial reference frame. The rotation matrices in Eq. (25) are used to transform from the

equatorial reference frame (EQ), to the payload module (PLM) reference frame, to the camera (CAM) boresight reference frame, to the (undistorted) focal plane (FP) reference frame, and finally to the CCD reference frame. The rotation matrix  $\mathbf{R}_{\text{EQ}}^{\text{PLM}}$  depends on three angles  $(\alpha_{\text{PLM}}, \delta_{\text{PLM}}, \kappa_{\text{PLM}})$  defining the orientation of the spacecraft, and  $\mathbf{R}_{\text{PLM}}^{\text{CAM}}$  depends on two angles  $(\eta_{\text{CAM}}, \rho_{\text{CAM}})$  defining the orientation of a camera on the payload module. All five of these angles are time dependent because of spacecraft pointing jitter and slow thermo-elastic drifts as explained in Sects. 6.3 and 6.4. The matrix  $\mathbf{R}_{\text{CAM}}^{\text{FP}}$  involves a pinhole projection as explained in Appendix A. The resulting focal plane coordinates are moreover subjected to optical distortion which is explained in Sect. 6.5.

### 6.3. Payload module pointing jitter

The AOCS controls the stability of the spacecraft pointing and is affected mainly by the reaction wheels and the fine guidance star system. As the AOCS is not perfect, the payload module jitters around a mean pointing, causing the stars to slightly move over the CCD (with a typical distance smaller than a pixel). The high-frequency components of the jitter cause PSF blurring, while the low-frequency components can displace the barycentre of the PSF from one pixel to the next. Due to the non-uniformity of the pixel-to-pixel response, the pointing jitter leads to increased photometric noise, and is thus a key driver for the photometric performance. Several correction algorithms have therefore been published so far in the literature (e.g. Drummond et al. 2006).

To model the pointing jitter we define the platform yaw  $\phi$ , the pitch  $\theta$ , and the roll  $\psi$  as rotation angles around respectively the  $X_{\text{PLM}}$ ,  $Y_{\text{PLM}}$ , and  $Z_{\text{PLM}}$  axes, such that the angles increase with a clockwise rotation, when looking along the positive axes. At any given time a perturbation to the pointing direction and the roll angle of the spacecraft is calculated by first performing a roll rotation around the  $Z_{\text{PLM}}$  axis, then a pitch rotation around the rotated  $Y_{\text{PLM}}$  axis, and finally a yaw rotation around the twice-rotated  $X_{\text{PLM}}$  axis. The combined rotation matrix (in the reference frame of the platform) is thus given by

$$\mathbf{R}(\theta, \varphi, \psi) = \mathbf{R}(\theta) \mathbf{R}(\varphi) \mathbf{R}(\psi). \quad (26)$$

In PlatoSim the update of the platform pointing using the rotation matrix above is done for every time step  $\delta t$  (cf. Eq. (6)).

The yaw, pitch, and roll time series used in the simulations are either taken from a detailed perturbation dynamical model of the spacecraft (not included in PlatoSim), or are simulated using red noise. In the latter case the jitter angles are modelled as in De Ridder et al. (2006), using a first-order auto-regressive model

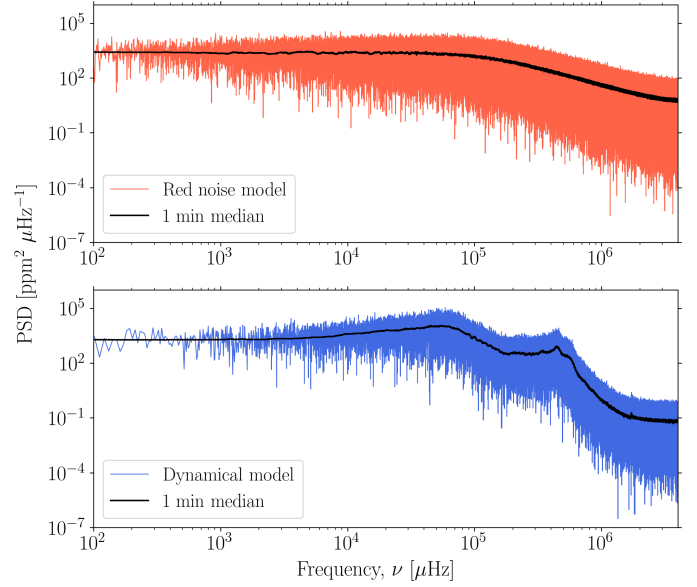
$$\theta_{n+1} = e^{-\delta t/\tau} \theta_n + \varepsilon_{n+1}. \quad (27)$$

Here  $\theta_n$  is the yaw angle a time  $t_n$ ,  $\tau$  is the jitter time scale,  $\delta t \ll \tau$  is the discretised time step, and  $\varepsilon$  is a Gaussian distributed noise fluctuation with zero mean and a variance equal to

$$\text{Var}[\varepsilon_n] = \sigma^2 \frac{\delta t}{\tau}, \quad (28)$$

where  $\sigma$  is the amplitude scale of the jitter.

Figure 8 shows for the yaw angle a power spectral density (PSD) function of the red noise model from PlatoSim (top panel using a rms amplitude scale of 0.04 arcsec) and dynamical model (bottom) produced by PLATO's prime contractor Otto



**Fig. 8.** Power spectral distribution (PSD) for a high frequency jitter time series of the yaw angle produced with PlatoSim's red noise model (top) and a dynamical OHB/TAS model (bottom). The two simulated models are sampled at 8 Hz and have a time duration of 27 h. The solid black line corresponds to a 1 min moving median filter.

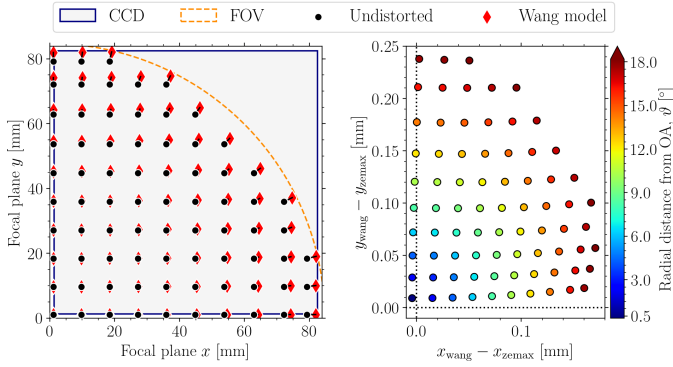
Hydraulic Bremen (OHB)/Thales Alenia Space (TAS). These simulations have a duration of 27 hours and a (fast) jitter time scale of 8 Hz. Compared to the dynamical model description of OHB/TAS, which assumes the instrument feedback from the F-CAMs to correct the pointing, the PSD proportionality  $1/f^2$  for the red noise model is lacking the correlated systematics between yaw, pitch, and roll (as seen by the additional kink of the OHB/TAS model). Nevertheless it has been shown that at a cadence of 25 s (i.e.  $10^4 \mu\text{Hz}$ ) the residual jitter noise of the dynamical model behaves Gaussian, which is, such as red noise, a stochastic process.

### 6.4. Thermo-elastic drift

To maintain the solar panels in the direction of the Sun, PLATO will perform a  $90^\circ$  rotation every three months after completing a quarter of the orbit around the Sun. During a run of three months, the spacecraft is not rotated to maintain a fixed field of view, which implies that the part of the spacecraft that is directly pointing towards the Sun gradually changes. In turn this means that the thermal profile of the spacecraft also slowly changes in time. In particular the thermal flexure of the optical bench will introduce slight changes to the pointing direction of each camera. This effect, also known as thermo-elastic drift (TED), will lead to a slow drift of the stars over the focal plane, up to 80% of a pixel as a worst case estimate for PLATO. The camera drift in PlatoSim is modelled using the same formalism as done for the AOCS jitter (see Sect. 6.3) taking the Euler angles (yaw, pitch, roll) as input either from a thermo-dynamical model or from a red noise model.

### 6.5. Optical field distortion

A last physical effect that impacts the positions of the star in the focal plane is the optical field distortion. In every real-life application a camera is subjected to image distortions due to slight manufacturing errors of the optical lens and relative optical



**Fig. 9.** Illustration of the field distortion models. Left: Distortion over one quadrant of the FPA with a grid step size of  $2^\circ$ . Shown are the undistorted paraxial chief ray coordinates (black dots) and the real (distorted) chief ray coordinates calculated by the Wang et al. (2008) distortion model (red diamonds), together with the CCD area (dark grey area enclosed by dark blue lines) and the effective size of the camera FOV (dashed orange line). Right: Residuals between the Wang and the Zemax distortion model evaluated in the FPA grid points shown in the left panel. The colour bar serves as a reference of the radial distance to the optical axis,  $\vartheta$ . We note that with a PLATO plate scale of  $18 \mu\text{m}$  a residual of 0.1 mm corresponds to  $\sim 5.6$  pixel.

alignment errors. Building from the heritage of the Brown-Conrady model (Brown 1971), a unified distortion model was formulated by Wang et al. (2008, referred to as the Wang model), which classifies the lens distortion into a radial, a tangential, and a thin prism distortion. Radial distortion is caused by an imperfect radial curvature of a lens, whereas the tangential (or decentring) distortion is caused by misalignments between different lens elements, and the thin prism distortion arises from a slight tilt of a lens with respect to the detector.

PlatoSim uses the self-contained distortion model in the case of the Zemax PSF, and implements the Wang distortion model for the analytic PSF. The Wang model is applied to all cameras and transforms the undistorted to the distorted (D) focal plane coordinates using

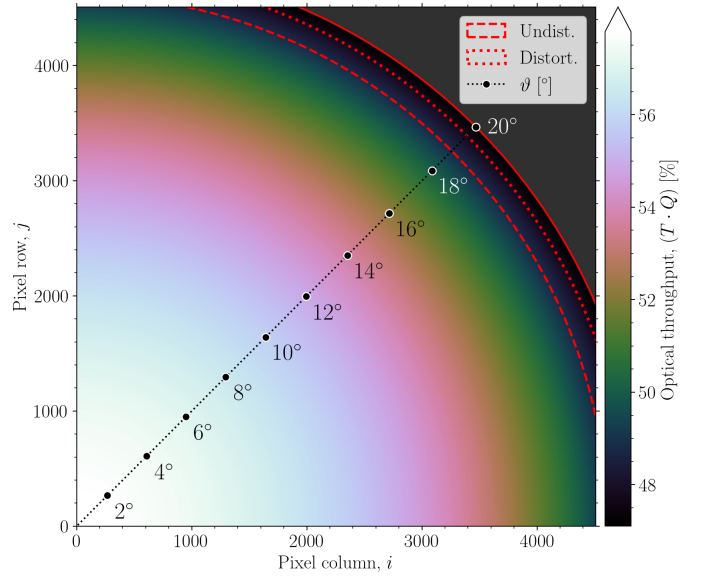
$$\begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \end{pmatrix}_D = \begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \end{pmatrix} + \begin{pmatrix} D_x \\ D_y \end{pmatrix}, \quad (29)$$

where

$$\begin{aligned} D_x &= x_{\text{FP}}(k_1 r^2 + k_2 r^4 + k_3 r^6) + x_{\text{FP}}(p_1 x_{\text{FP}} + p_2 y_{\text{FP}}) + q_1 r^2, \\ D_y &= y_{\text{FP}}(k_1 r^2 + k_2 r^4 + k_3 r^6) + y_{\text{FP}}(p_1 x_{\text{FP}} + p_2 y_{\text{FP}}) + q_2 r^2. \end{aligned} \quad (30)$$

Here  $r = \sqrt{x_{\text{FP}}^2 + y_{\text{FP}}^2}/l$  is the undistorted radial distance from the optical axis normalised by the focal length  $l$ . The set of coefficients  $(k_1, k_2, k_3, q_1, q_2, p_1, p_2)$  belongs to the model description of respectively the radial ( $k$ ), the tangential ( $p$ ), and thin prism ( $q$ ) component. The coefficients used by PlatoSim to model the direct and inverse distortion have been derived using a Zemax model as part of the mission preparation. In addition, PlatoSim allows the coefficients to be provided as a time series, to allow modelling the effect of a changing thermal environment on the optical distortion.

An illustration of the expected field distortion model representative for the PLATO camera is shown in the left-hand panel of Fig. 9 for one quadrant of the focal plane. Here the black dots represent the undistorted (i.e. unobservable distortion-free) chief ray positions and the red diamonds are the distorted



**Fig. 10.** Illustration of the total throughput map for one full frame CCD. The dotted diagonal line shows the distance from the optical axis in degrees ( $\vartheta$ ) and the red dashed lines show the angular position of the stray light mask. We note that the FOV in the focal plane physically extends beyond  $\vartheta_{\text{max}}$  (to  $\sim 19.6^\circ$  indicated by the red dotted line) due to the effect of optical distortion cf. Sect. 6.5 and is followed by an exponential intensity decay of vignetting (modelled out to  $20^\circ$  shown by the red solid line).

chief ray positions from the Wang model. The right-hand panel of the same figure shows the residual plot between the Zemax and Wang model computed at the same FOV grid shown in the left-hand plot. Overall the residual plot shows a good agreement between the two models, with the Wang model generally resulting in a slightly stronger distortion compared to the Zemax prediction, and where the largest discrepancies are farthest from the optical axis (dashed orange line in the left-hand plot).

## 7. Optical throughput and detector efficiency

This section describes the throughput and efficiency quantities  $\bar{T}(t, x, y)$ ,  $\bar{E}(t, x, y)$ , and  $\bar{Q}(t, x, y)$  that occur in Eq. (8).

### 7.1. Optical throughput

The total optical throughput (also known as the spectral response) of an optical instrument represents its efficiency to convert incident photons into counts of electrons at detector level. Hence it is a product of the dimensionless optical transmission  $T_{ij}(t, \lambda)$  and the quantum efficiency  $Q_{ij}(t, \lambda)$ . The time-dependence comes from a slow degradation between the beginning and the end of the mission (BOL  $\rightarrow$  EOL), which is modelled in PlatoSim using a linear relation. We discuss this model choice in the following. Figure 10 illustrates the total optical throughput integrated over the PLATO passband, at the beginning of life over one full-frame CCD.

The monochromatic transmission  $T_{ij}(t, \lambda)$  combines the effects of the transmission efficiency of a photometric pass-band filter  $T_{\text{fil}}$ , the vignetting  $T_{\text{vin}}$ , the particulate and molecular contamination  $T_{\text{con}}$ , and the polarisation transmission efficiency  $T_{\text{pol}}$ ,

$$T_{ij}(t, \lambda) = T_{\text{fil}}(t, \lambda) T_{\text{vin},ij}(\vartheta, \lambda) T_{\text{con}}(t, \lambda) T_{\text{pol},ij}(\vartheta, \lambda). \quad (31)$$

The polychromatic version, as appearing in Eq. (8) and used by PlatoSim, is derived by taking the average over the PLATO passband

$$\bar{T}_{ij}(t) = \bar{T}_{\text{fil}}(t) \bar{T}_{\text{vin},ij}(\vartheta) \bar{T}_{\text{con}}(t) \bar{T}_{\text{pol},ij}(\vartheta). \quad (32)$$

The phenomenon of vignetting, that is the brightness attenuation towards the edge of the FOV, can be divided in three components: natural, optical, and mechanical vignetting. The attenuation by natural vignetting is caused by the fact that off-axis light rays not only have a longer travel distance but also see a projected (i.e. reduced) area of the entrance pupil, leading to a decreasing light intensity at angles far away from the optical axis. Optical vignetting is induced by the optical design of a camera that features multiple optical elements. A lens earlier in the light path causes a reduction of the effective opening of the next lens because the output angles of the former are limited. Also this causes a decrease in intensity towards the edge of the FOV. Mechanical vignetting is due to the blocking of light rays by the straylight mask, and causes a semi-hard circular border of the FOV at a maximum angle  $\vartheta_{\text{max}}$  from the optical axis. The vignetting for a PLATO camera was analysed using a Zemax optical model, as well as measured during the camera assembly, integration, and verification, which led to the following best fitting parametric model

$$\bar{T}_{\text{vin}}(\vartheta) = \begin{cases} 1 - k_1\vartheta^2 - k_2\vartheta^4 - k_3\vartheta^6 & \text{for } \vartheta \leq \vartheta_{\text{max}} = 18.9^\circ, \\ c + e^{-(\vartheta - \vartheta_{\text{max}})/\sigma} & \text{for } \vartheta > \vartheta_{\text{max}}, \end{cases} \quad (33)$$

where  $\{k_1, k_2, k_3\} = \{4.18 \cdot 10^{-2}, -5.65 \cdot 10^{-5}, 2.37 \cdot 10^{-7}\}$ . Here,  $\sigma = 0.6^\circ$  and  $c$  is a constant so that the function is continuous in  $\vartheta = \vartheta_{\text{max}}$ .

The particulate contamination is the (unintended) presence of particles on (mostly optical) surfaces, whereas the molecular contamination is the layer of molecules on top of a surface caused by out-gassing of materials in the first phase of the mission (for an in depth discussion see e.g. Zhao et al. 2009). A major part of the particulate contamination takes place during the fairing ejection (i.e. well before PLATO will start its journey to the L2) which, by and large, sets the level of contamination in the camera entrance window. The story of the molecular contamination is on the other hand more complex. The majority of out-gassing typically takes place during the cooldown of the spacecraft (being mostly during the first three days after launch), but will in reality never stop completely. Furthermore, the out-gassing can also take place from various materials at different rates. However, we assume that out-gassing during launch is neglectable for PLATO, due to the spacecraft's limited duration within Earth's atmosphere, together with a slow rate of change for out-gassing at the time PLATO starts operating in the L2. Hence, as mentioned before, a linearly decreasing model of the transmission efficiency is a good approximation if radiation damage is the dominating factor for the degradation. PlatoSim uses a throughput value of 0.972 due to particulate contamination and a value of 0.9573 due to molecular contamination, which are the BOL requirement values.

The polarisation transmission efficiency is modelled using the following (fairly arbitrary monotonically decreasing) parametric model

$$\bar{T}_{\text{pol},ij}(\vartheta) = \bar{T}_{\text{pol,max}} \cos\left(\frac{\vartheta}{\vartheta_{\text{ref}}} \cos^{-1}\left[\bar{T}_{\text{pol}}(\vartheta_{\text{ref}})\right]\right), \quad (34)$$

where  $\bar{T}_{\text{pol,max}}$  is the  $\bar{T}_{\text{pol}}$  maximal value and  $\bar{T}_{\text{pol}}(\vartheta_{\text{ref}})$  is the value at a certain reference angular distance  $\vartheta_{\text{ref}}$  away from the optical axis.

Lastly, the quantum efficiency (QE) first presented in Eq. (6) is generally defined as

$$Q_{ij}(t, \lambda) = \frac{Q_{\text{ext},ij}(t, \lambda)}{1 - T_{\text{ref},ij}(t, \lambda)}, \quad (35)$$

where  $Q_{\text{ext}}$  is the external quantum efficiency, that is the ratio of the number of electrons over the number of incident photons, and  $T_{\text{ref}}$  is the reflectivity, that is the fraction of photons reflected at the surface despite the anti-reflection coating. This expression implies that the quantum efficiency also depends to second order on the angle of the incident light. Since the PLATO payload's optical design leads to CCD illumination over a wide range of incident angles (up to  $40^\circ$ ; Rauer et al. 2014, and in prep.), the variation of the QE with incidence angle is taken into account in PlatoSim. As before we avoid dealing with the wavelength dependence of the QE by averaging it over the passband  $\bar{Q}_{ij} \approx \langle Q_{ij}(\lambda) \rangle_\lambda$ , and we use a similar parametric model as above

$$\bar{Q}(\vartheta) = \bar{Q}_{\text{max}} \cos\left(\frac{\vartheta}{\vartheta_{\text{ref}}} \cos^{-1}\left[\bar{Q}(\vartheta_{\text{ref}})\right]\right). \quad (36)$$

## 7.2. Detector efficiency

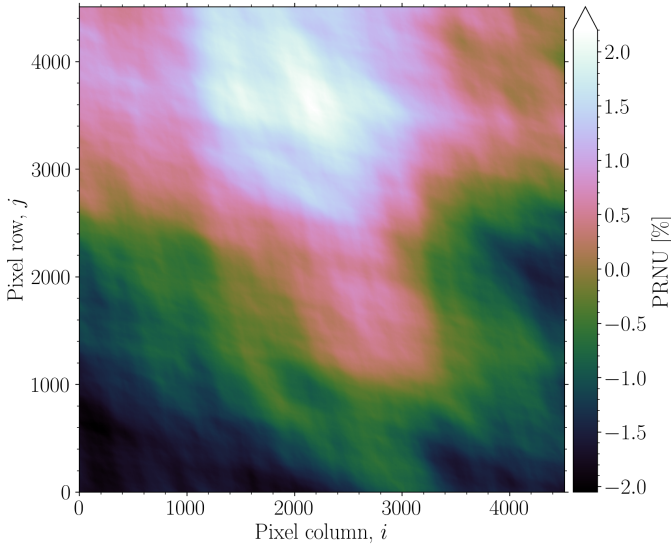
The detector efficiency  $\bar{E}_{ij}(t)$  in Eq. (8) encompasses both spatial pixel sensitivity variation as well as defective pixels. The former is caused by the fact that the electric field structure within a pixel is affected by small variations in pixel size, the structure of the gate electrodes, the thickness of the SiO<sub>2</sub> insulation layer, and the doping uniformity in the epitaxial (crystalline) Si layers (see e.g. Jorden et al. 1994). The combination of pixel sensitivity variations and spacecraft pointing jitter introduces tiny flux variations that increase the noise level. PlatoSim therefore includes the pixel-response non-uniformity (PRNU; often referred to as the flat-field for space-borne instruments) in its simulations.

The most reliable flat-field is an empirical one obtained from measuring the PRNU at different wavelengths using a flight model of the camera. These measurements are then combined in a weighted average over the PLATO passband. Early in the design of the space mission such measurements are not available, in which case PlatoSim resorts to a simulated flat-field. The important feature here is that the spatial variation in pixel sensitivities usually do not follow a white noise pattern, but shows a spatial correlation. To simulate this, PlatoSim first models the 2D Fourier transform  $\bar{E}_{\text{FT}}(m, n)$  of the PRNU map as

$$\bar{E}_{\text{FT}}(m, n) = \frac{\varepsilon_{m,n}}{1 + m^\beta + n^\beta}, \quad (37)$$

where  $(m, n)$  are the spatial wavelengths,  $\varepsilon_{m,n} \sim \mathcal{N}(0, 1)$  is Gaussian distributed noise fluctuation, and the exponent  $\beta$  determines the strength of the correlation at larger spatial distances on the CCD. Subsequently, the inverse Fourier transform is taken and scaled to have a given mean and rms, to obtain the actual PRNU map  $\bar{E}(i, j)$ . Figure 11 shows an example of a relative flat-field for  $\beta = 2$ , which shows a strong spatial correlation over the CCD.

Defective pixels also impact the detector efficiency and fall under one of the following three categories: dead, hot, or telegraphic. Compared to a normal pixel a dead pixel has an anomalously low sensitivity, whereas a hot pixel has an anomalously high dark current. A telegraphic (or RTS; random telegraph



**Fig. 11.** Illustration of the automatically generated flat-field (PRNU) for a full-frame CCD image. This image represents the flat-field used to construct the subfield in Fig. 12 and has a peak-to-peak pixel sensitivity variation  $\sim 4\%$  and a local rms noise level of  $\sim 1\%$ .

signal) pixel, on the other hand, is a pixel which periodically switches between an active state with high dark current and an inactive state with normal dark current, and vice versa. While the identification of defective pixels has been determined as part of the on-ground calibration on a flight model CCD (Verhoeve et al. 2016), the relative fraction of defective pixels is less than 0.1% and are thus not expected to pose a problem for PLATO. Therefore, defective pixels are currently not implemented in PlatoSim, however, we do acknowledge the potential importance of implementing these (in particular RTS pixels) later in the mission.

## 8. Electron redistribution models

### 8.1. Brighter-fatter effect

The brighter-fatter effect (BFE) is an electron redistributing effect caused by electrostatic interaction between neighbouring pixels. Pixels that already collected a large number of electrons during an exposure shrink in effective collecting area, so that they collect and retain less electrons because some of them are now attracted to a neighbouring pixel. The BFE is different from charge diffusion discussed in Sect. 5.1 in the sense that the former is caused by a changing electrostatic field configuration during an exposure, while the latter is caused by electrons drifting laterally during a random walk before ending up in a pixel. The BFE phenomenon is fairly well understood, and the model implemented in PlatoSim uses the approximative framework outlined by Antilogus et al. (2014), Guyonnet et al. (2015) and Astier et al. (2019).

The electrostatic field lines caused by the pixel gate electrodes define the path of the incoming electrons and therefore determines to which pixel the electron is attracted. In the case of a perfect regular grid of electrodes, the pixel boundaries fall at the geometrical midpoint between two electrodes as is nicely illustrated in Fig. 4 of Antilogus et al. (2014). When a pixel accumulates electrons the electrostatic field lines change as the accumulated electrons have a repulsive effect on new incoming electrons. Electrons close to the geometrical midpoint are

now attracted more by the neighbouring electrode, effectively shifting the pixel boundary towards the first electrode so that its corresponding collecting area becomes smaller. The boundary between a ‘central’ pixel ( $C$ ) and its neighbouring pixel ‘north’ ( $N$ ) can be affected by the number of electrons in a nearby (although not necessarily neighbouring) pixel ‘ $P$ ’. In a first approximation the boundary shift  $\delta X^{C \leftrightarrow N}$  scales linearly with the charge  $Q_P$  already accumulated in pixel  $P$ . Since the electric field is additive, the total effect can be computed by summing up the effects of all nearby pixels  $\{P\}$

$$\delta X^{C \leftrightarrow N} = \frac{1}{2} \sum_{\{P\}} a_P^{C \leftrightarrow N} Q_P \quad (38)$$

where the factor  $1/2$  is added to be compatible with the definition of the coefficients  $a_P^{C \leftrightarrow N}$  as defined in Guyonnet et al. (2015). The sum of the linear coefficients  $a_P^{C \leftrightarrow N}$  should be zero as we do not expect any net shift to happen when all charges  $Q_P$  are equal.

Antilogus et al. (2014) argue that the change in charge  $\delta Q^{C \leftrightarrow N}$  in pixel  $C$  due to the boundary shift  $C \leftrightarrow N$  scales in a first approximation linearly with the shift  $\delta X^{C \leftrightarrow N}$  as well as with the charge density at the boundary between the two pixels which scales to first order by the total amount of charge  $Q_C + Q_N$

$$\delta Q^{C \leftrightarrow N} = \frac{1}{4} \sum_{\{P\}} a_P^{C \leftrightarrow N} Q_P (Q_C + Q_N), \quad (39)$$

where we added an extra scale factor  $1/2$  instead of absorbing it in the coefficients  $a_P^{C \leftrightarrow N}$  to be compatible with Guyonnet et al. (2015). The coefficients  $a_P^{C \leftrightarrow N}$  in Eq. (39) still sum up to zero. The change in charge in pixel  $C$  is also affected by the change in boundary with the other neighbouring pixels ‘south’ ( $S$ ), ‘west’ ( $W$ ), and ‘east’ ( $E$ ), so that we can write

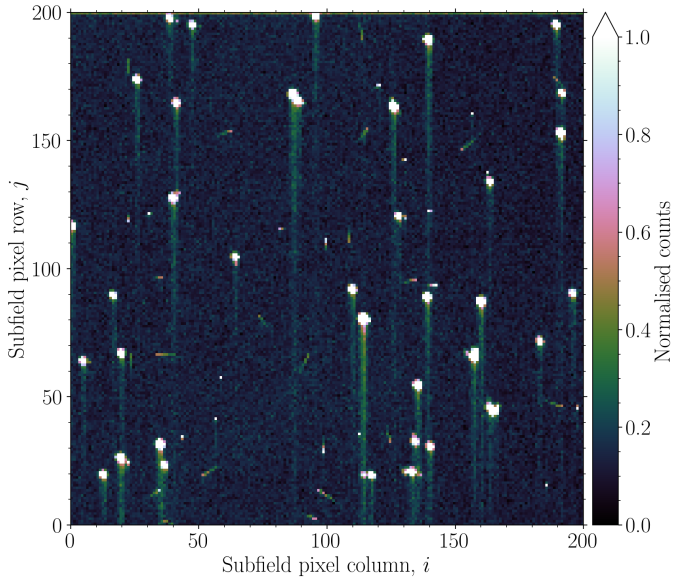
$$\begin{aligned} \delta Q_C &= \delta Q^{C \leftrightarrow N} + \delta Q^{C \leftrightarrow S} + \delta Q^{C \leftrightarrow E} + \delta Q^{C \leftrightarrow W} \\ &= \frac{1}{4} \sum_{\{X\}} \sum_{\{P\}} a_P^{C \leftrightarrow X} Q_P (Q_C + Q_X) \end{aligned} \quad (40)$$

where  $\{X\}$  stands for {north, south, west, east}. The coefficients  $a_P^{C \leftrightarrow X}$  were derived in two steps. First, the inter-pixel variance and covariance curves were computed using a set of flatfields with a large range of fluxes, using the prescription of Astier et al. (2019). Then, the electrostatic model of Astier & Regnault (2023) was fitted to obtain the coefficients  $a_P^{C \leftrightarrow X}$ .

### 8.2. Charge transfer efficiency

The effect of charge-transfer inefficiency (CTI) happens during the readout of the CCD because of imperfections in the CCD silicon substrate lattice, which creates electron traps. Due to the stochastic capture and release of electrons into and out of these traps, part of the signal is left behind when some electrons are trapped while being transferred during readout. The delayed release of electrons is subsequently causing a smearing effect leading to the well-known CTI tails. As radiation damage is the leading cause for the creation of these traps, the CTI deteriorates fast with time for any space mission due to the increased radiation dose received compared to that of ground-based instruments (e.g. Massey et al. 2014).

A first simple CTI model implemented in PlatoSim follows the fraction of the total charge in a pixel  $\theta_{CTE}$  that is left behind,



**Fig. 12.** Illustration of the effect of CTI at post mission EOL (i.e. 6.5 yr after commissioning) using the Short model. The plot shows a centrally placed  $200 \times 200$  pixel CCD subfields of PIC stars from the LOP south including cosmic rays simulated using a hit rate of  $10 \text{ events s}^{-1} \text{ cm}^{-1}$ . The images has been clipped by a  $2\sigma$  cut and then normalised for illustrative purposes. The readout register is located towards the bottom of the image.

hence,  $\theta_{\text{CTI}} = 1 - \theta_{\text{CTE}}$ . As charge transfer happens both during parallel transfer as well as serial transfer, the charge that is lost during each of these transfers is (Janesick 2001)

$$Q_{N+n} = \frac{Q_0 N!}{(N-n)! n!} (1 - \theta_{\text{CTI}})^n \theta_{\text{CTI}}^{N+n}, \quad (41)$$

where  $Q_0$  is the initial charge contained in the target pixel,  $N$  is the number of pixel transfers,  $n$  is the trailing pixel number following the target pixel (with  $n = 0$  being the target pixel itself), and  $Q_{N+n}$  is the remaining charge in the  $N + n$  pixel.

As a more physical description (naturally introducing the time dependence), the analytic model proposed by Short et al. (2013); abbreviated the Short model) for radiation-induced CTI for CCD detectors is also implemented into PLATOsim. This implementation includes 4 different CTE trap species, each with its own trap density  $n_t$  (traps  $\text{pixel}^{-1}$ ), trap capture cross-section  $\sigma_t$  ( $\text{m}^2$ ), and a release time scale  $t_t$  (s). In addition, the model also requires a parameter  $\beta$  describing whether adding electrons to a charge package is increasing the volume of the package or its density. Prod'homme et al. (2016) derived BOL and EOL values for these quantities for the PLATO mission, assuming a nominal operating temperature of 203 K. Figure 12 illustrates a worst case example of the effect of CTI at EOL (6.5 yr after commissioning) using the Short model. While the CTI increases with time the photometric quality decrease correspondingly, hence, the PLATO data reduction pipeline therefore contains a CTI correction step (see e.g. Israel et al. 2015, for good overview of such corrections).

### 8.3. Blooming

For bright stars ( $V \leq 8.5$  for N-CAM observations) the full-well capacity of some of the pixels is reached, and the electrons start to overflow to neighbouring pixels in the same column,

which is usually referred to as blooming. Although this saturation happens during the CCD exposure, PLATOsim models it pragmatically as a post-exposure effect that is independent of CCD non-linearity (which is applied later). The caveat of this implementation results in a slightly enhanced blooming pattern at the expense of computational speed, compared to an iterative approach which models their interaction every time step  $\delta t$ . In a first model, the electron excess is simply distributed evenly between the pixels above and below the saturated pixel. If these pixels also get saturated, the overflow goes to the next pixel in the column, etc. The result is a blooming pattern that is symmetric with respect to the central pixel of the star, that is the upward blooming trail has the same length as the downward one.

Tests with the PLATO CCDs revealed, however, that in practice blooming can be highly asymmetric, with one trail being significantly longer than the other one. Although the exact underlying cause is still being investigated, the working hypothesis is that the electrons experience barriers in one direction, so that the excess electrons follow the path of less resistance in the opposite direction. Rather than trying to mimic the still uncertain physical causes, PLATOsim simply allows to specify the fraction of excess electrons that goes downwards (i.e. towards the readout register), which is sufficient to test the design and testing of the mask creation and photometry extraction of saturated stars.

## 9. Photometry

One of the main applications of PLATOsim is its ability to provide realistic light curves extracted at pixel level using its build-in photometric algorithm. Besides alleviating PLATOsim's ability to generate light curves on demand, this feature is especially important when running large batches of simulations where the raw pixel data and house keeping data may be of the order of several gigabytes (or even terabytes). Due to the new software design of PLATOsim, generating light curves at run time significantly reduces the storage memory for output and adds a minimal time of execution per simulation.

To date a noteworthy list of literature exists on software for PSF photometry (Da Costa 1992; Schechter et al. 1993; Anderson & King 2006; Popowicz 2018; Hedges et al. 2021), aperture photometry (Howell 1989; Stetson 1987; Naylor 1998; Libralato et al. 2016; Bryson et al. 2010; Handberg & Lund 2014; Lund et al. 2015; Aigrain et al. 2016; Smith et al. 2016; Marchiori et al. 2019; Hoyer et al. 2020), or pipelines that either combine both or provide both methodologies in a single software package (Kjeldsen & Frandsen 1992; Still & Barclay 2012; Bradley et al. 2016; Lightkurve Collaboration et al. 2018). Generally the PLATO light curve generation of non-saturated stars follows two separate data processing chains, namely PSF photometry designed for on-ground data products (i.e. imagettes of  $V < 11$  stars belonging to the asPIC sample called P1) and aperture photometry designed for on-board data products (i.e. flux measurements of fainter targets ( $11 < V < 15$ ) primary consisting of stars from the asPIC sample called P5). For now only the a subset of the on-board algorithms are implemented in PLATOsim whereas a functional coupling to the full processing chains, on-board and on-ground, have been established as will be explained in Sect. 11.2.

The performance reached by the photometry (both on-ground and on-board) directly depends on the knowledge of the PSF across the focal plane for each independent pointing. However, since the PSF morphology is expected to change notably after launch (due to slight changes of the optical mount during launch and to changes in the thermal environment throughout the



mission) the ‘true’ PSF cannot be measured from ground. Furthermore, acquiring accurate knowledge about the PLATO PSF in-flight is a main challenge due to the sparse pixel sampling of the PSF. Thus, we first elaborate on the procedure to overcome this challenge by reconstructing the true but unknown PSF across the CCD focal plane.

### 9.1. PSF inversion using microscanning

PSF reconstruction builds from the idea of extracting a high resolution PSF,  $\mathbf{x}$ , from a series of corresponding lower resolution PSFs,  $\mathbf{y}$ , following (Park et al. 2003)

$$\mathbf{A} \mathbf{x} = \mathbf{y}, \quad (42)$$

where  $\mathbf{A}$  is a PSF projection matrix onto the low-resolution pixel grid. Mathematically the inversion is solved by discretising the PSF using a sum of basic functions  $\phi_i$

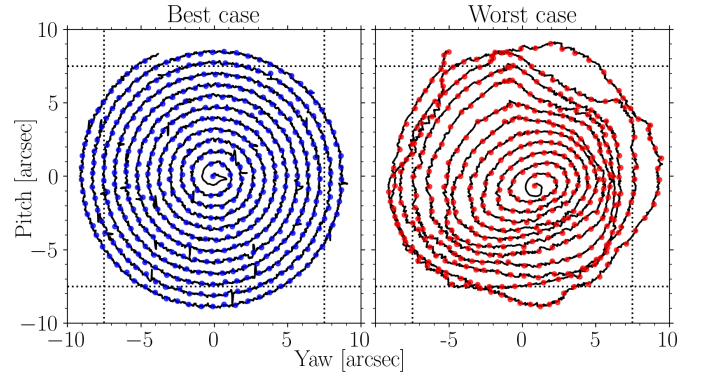
$$f(x, y) = \sum_i a_i \phi_i(x, y), \quad (43)$$

with  $a_i$  being the unknown inversion coefficients. Solving Eq. (42) can be tackled using a least-squares procedure. The preferred method for PLATO is discussed in Samadi et al. (2019).

Following from the PSF inversion technique first employed by CoRoT (Auvergne et al. 2009) and later matured with Kepler (Bryson et al. 2010), in practice the series of low resolution pixel frames,  $\mathbf{y}$ , is acquired by intentionally commanding the AOCS to follow a certain pattern of small coordinate displacements. Such selective jittering is also known as a microscanning session. Figure 13 displays the Archimedean spiral pattern decided for PLATO, shown for a best case (left) and worst case (right) scenario in terms of pointing stability during the session.

Microscanning sessions will be performed during the quarterly interruptions necessary to realign the spacecraft’s solar panels and will have a duration of around three hours each. The current in-flight strategy for microscanning is going to provide the inverted PSF for all P1 sample stars, whereas only a subset of carefully selected P5 (named R2) sample stars across the FOV will have their PSFs inverted for each pointing. Thus, the remaining P5 targets will have their high resolution PSFs determined from interpolation. The exact scheme of interpolation and for which magnitude range inverted PSFs reliably can be determined has been established by the PLATO team responsible for the data processing and algorithms. A transition to this interpolation strategy is currently being integrated into PlatoSim. As an approximate but realistic approach, PlatoSim allows to use a precomputed grid of inverted PSFs generated from both the worst and best case microscanning examples displayed in Fig. 13.

Currently the grid of inverted PSFs are only fully representative for a single camera and quarter. In practice a grid of inverted PSFs for each camera is ideal, since the PSFs are strongly dependent on the alignment of optical lenses in each TOU, and for each mission quarter, since changes in PSF morphology together with ageing effects (such as the CTI) impact the accuracy of the inversion increasingly over time. Thus, particularly important for EOL conditions, more microscans are needed in the future to bring the photometry closer to the current mission strategy. However, as the state-of-the-art PSF- and aperture photometry algorithms require a high resolution PSF as input, this approach is already more realistic than the unphysical use of the ‘true’ PSF (be it Zemax or analytical).



**Fig. 13.** Illustration of the Archimedean spiral jitter pattern including; Left: Minor residual AOCS jitter as a best case. Right: Major AOCS residual jitter as a worst case. For an ideal pattern the distance  $D$  between consecutive measurements is approximately constant and distance between consecutive spiral arms is  $D\sqrt{3}/2$ . These files contain 430 scans (shown as coloured circles marking the start of each exposure) and forms a near-equilateral triangular grid, which in turn provides a dense but near complete sampling over PLATO’s pixel grid (dotted lines), needed for a successful PSF inversion. Data courtesy: OHB/TAS.

### 9.2. Pre-processing steps

Prior to the photometry extraction several pre-processing steps are applied to the simulated CCD subfield. First the bias offset is subtracted by computing the mean over a serial prescan region (orange box in Fig. 3a) and a virtual<sup>9</sup> overscan region from either the F or E side of the detector, dependent on the subfield location. Next, readout smearing is corrected for by subtracting the bias corrected smearing map obtained from the parallel overscan region (pink box in Fig. 3a). The gain is then used to convert the pixel values from ADU to counts of photoelectrons. Lastly, the bias subtracted sky background map is multiplied with the overall throughput to get counts of  $e^- \text{pixel}^{-1} \text{exposure}^{-1}$  and then subtracted from the pixel map.

### 9.3. Optimal aperture algorithm

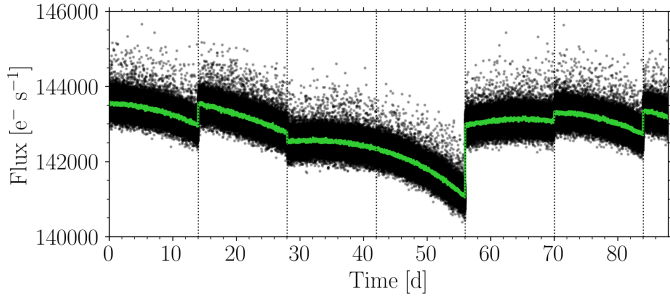
From the wealth of aperture photometry pipelines mentioned above the on-board algorithm implemented in PlatoSim follows from, a study by Marchiori et al. (2019). By design, it optimises the photometric quality towards planet transit searches. This study found that a binary mask serves as the best compromise between noise-to-signal ratio (NSR) and the ratio of stellar contamination.

The general idea is to build a mask starting with the pixel having the lowest NSR, then adding one pixel at a time under the condition that adding it should contribute more to the aggregated signal than to the aggregated noise. For an imagette, this procedure can be formulated mathematically by first arranging all  $n$  pixels in increasing order of NSR

$$\text{NSR}_n = \frac{\sqrt{\sigma_{F_{T_n}} + \sum_{k=1}^{N_c} \sigma_{F_{C_{n,k}}} + \sigma_{B_n} + \sigma_{D_n} + \sigma_{Q_n}}}{F_{T_n}}, \quad (44)$$

where  $F_{T_n}$  and  $\sigma_{F_{T_n}}$  are respectively the mean flux and photon noise of the target star,  $\sigma_{F_{C_{n,k}}}$  is the photon noise of each  $k$  stellar contaminant,  $\sigma_{B_n}$  is the sky background noise,  $\sigma_{D_n}$  is the combined detector noise, and  $\sigma_{Q_n}$  is the quantisation noise.

<sup>9</sup> Virtual as in extra readouts of the register.



**Fig. 14.** On-board photometry performed with the optimal aperture algorithm of Marchiori et al. (2019) for a  $V = 10$  star. With a central barycentric pixel position and a (worst case) systematic drift of 1.3 pixel over the course of one mission quarter, the figure shows the algorithm in action with the automatic pixel-mask updates triggered every 14 days (grey-dotted lines) if a lower NSR can be achieved (which is not the case for the update at 42 days). Since the NSR does not scale linearly with flux, the mask-update strategy does not necessarily increase the flux level (as the case at 28 days). The positive flux outliers are due to contamination from cosmic rays (with a hit rate of  $10 \text{ events s}^{-1} \text{ cm}^{-2}$ ).

The second step consists in determining the aggregated NSR over the imagette’s  $m$  pixels conforming to the aforementioned pixel order of increasing NSR

$$\text{NSR}_{\text{agg}}(m) = \frac{\sqrt{\sum_{n=1}^m (\sigma_{F_{T_n}} + \sum_{k=1}^{N_c} \sigma_{F_{C_{n,k}}} + \sigma_{B_n} + \sigma_{D_n} + \sigma_{Q_n})}}{\sum_{n=1}^m F_{T_n}}. \quad (45)$$

The last step is simply to construct the aperture from the collection of pixels  $m$  that minimises Eq. (45).

Pointing performance degradations over long time scales (due to long-term drifts) and on short time scales (due to reaction wheel momentum dumps, attitude tweaks, loss of fine guidance, pre/post safe mode events, etc.) may introduce significant pixel displacements. Compared to PSF photometry which by design is more robust against such instrumental perturbations, for aperture photometry this lead to systematic errors in the form of flux loss outside the pixel mask. To mitigate the loss of photometric precision over the course of a mission quarter the strategy for PLATO is to update the aperture of each star periodically under the condition that a lower NSR can be achieved by the updated mask (Marchiori et al. 2019).

Investigations led by the PLATO performance team have shown that the combined effect of TED and KDA can lead to a barycentric displacement up to 1.3 pixel over a three months duration. We illustrate this worst case scenario in Fig. 14 for a  $V = 10$  star positioned  $5^\circ$  from the optical axis at an initial central intra-pixel position. As the star drifts over the pixel array the mask update events (grey dotted lines) manifest themselves as flux jumps (which is amplified with a 1 h running flux median shown in green in Fig. 14). This highlights a strategy trade-off between the mask update frequency vs. potential error propagation from the pipeline corrections. To minimise the mask update frequency developers of the PLATO pipeline are currently investigating if suboptimal apertures can accommodate the fully predictable KDA contribution on the stellar barycentric displacement across the FPA. For further information we refer to Samadi et al. (2019) for a discussion on the impact of the mask updates on the final light curve and how long term and short term pixel displacements can be corrected in the post-processing procedure.

It is noteworthy that systematic noise sources acting on time scales shorter than the cadence are not included in the mask creation of Eq. (45). This is the case for AOCS jitter and to lesser extent CCD effects that degrade the photometric quality (such as CTI, cosmetic defects, etc.). Including the contribution of jitter (and larger attitude tweaks) is a challenge since it depends on the final shape of the mask. Nevertheless, instrumental perturbations such as jitter has been shown to have a negligible impact on the photometry in nominal conditions (Marchiori et al. 2019), and can partially be corrected for during the PLATO pipeline chain (cf. Samadi et al. 2019). Moreover, as soon as the aperture mask of Eq. (45) has been defined, the stellar pollution ratio (SPR) from nearby contaminant can be derived, which can help for rejecting false-positive planet detections from blended eclipsing binaries.

## 10. PlatoSim software architecture

The creation of a synthetic CCD image starts with a set of input parameters that defines the general properties of the spacecraft’s hardware components, the stellar field, and the simulated observation itself. As depicted in Fig. 15 the PlatoSim software generates a simulation in two steps: a) it configures all input parameters and constructs an output file and then b) it follows all (requested) algorithm steps in a loop over a total number of exposures defined by the user.

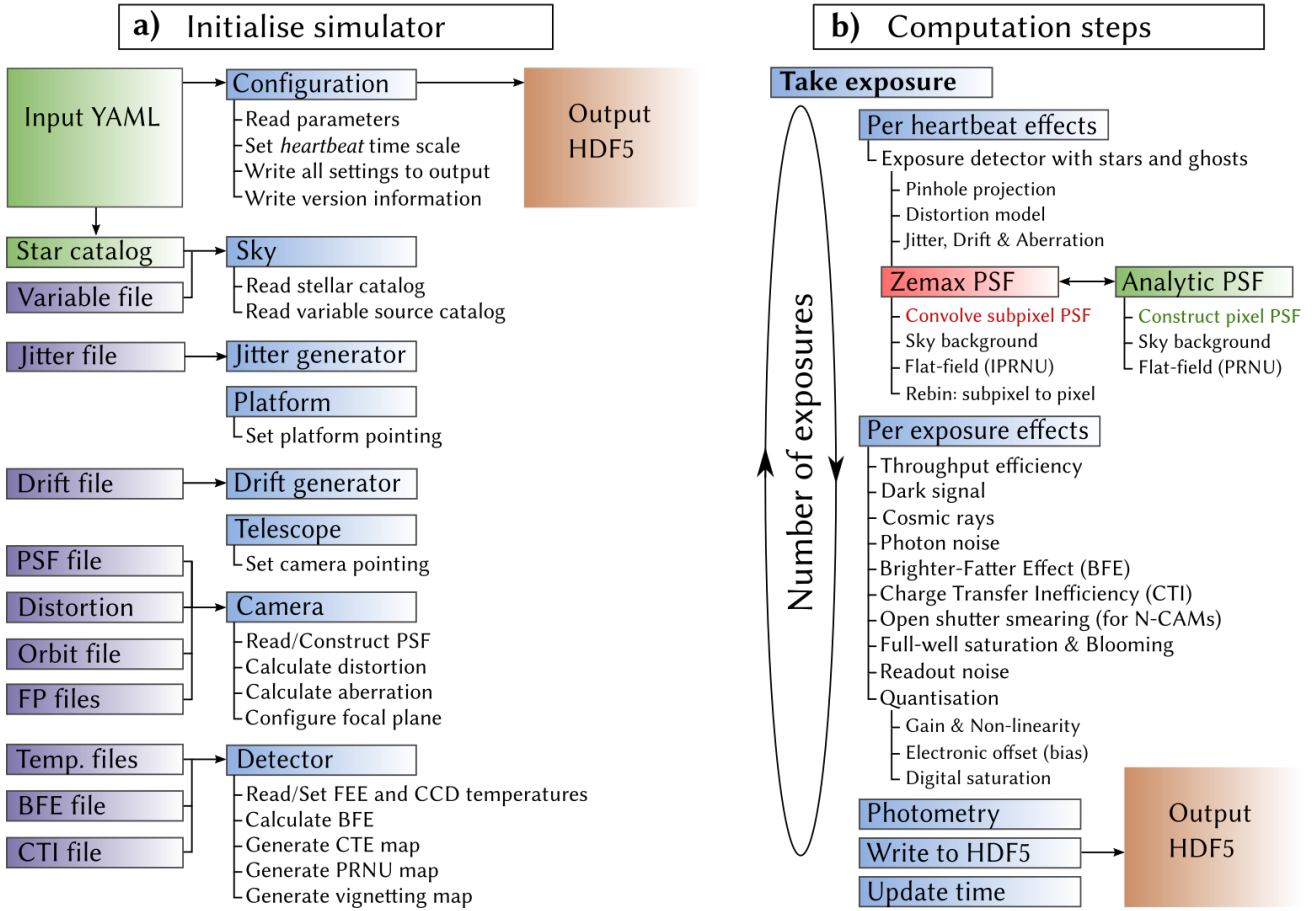
The simulation construction relies on a YAML configuration file that initialises all the input parameters and optionally reads further supplementary files (purple boxes). As a minimum PlatoSim needs a YAML input file and a star catalogue to successfully run (green boxes). Identical to the physical hardware components that make up the PLATO payload, PlatoSim consist of a platform, telescope<sup>10</sup>, camera, and detector module, in combination with a sky module to include sky background and stellar variable signals, and two time series generators (jitter and drift) for the inclusion of pointing systematics in an automated way (blue boxes). Each module deals with a particular effect or subsystem. Combined, these are controlled by a global simulation object that is directly configurable in Python. As mentioned in Sect. 4, the smallest time varying phenomena, referred to as PlatoSim’s ‘heartbeat’, is initialised in order to partition each exposure into smaller time steps.

The construction of a simulation with respect to input and output is completely standardised in modules. This is highly beneficial for understanding subsystems or individual effects. All supplementary input files should be provided in ascii format and output files are in HDF5 format. As indicated by the purple boxes in Fig. 15a, whether an effect is included and/or a model selected leaves an enormous flexibility for the user to conduct highly diverse (and complex) simulations.

Upon execution, after the setup has finished in step a), PlatoSim generates a time series of synthetic CCD images in a loop over the total number of requested exposures, as illustrated in Fig. 15b. Over the course of a single exposure, the algorithmic steps are organised into two consecutive classes: first those that are computed per heartbeat, and next those that are computed only once per exposure. The flow of the algorithmic steps are approximately according to the light path of the incident photons, placing each effect or subsystem in a logical order after their physical occurrence. However, as mentioned earlier this is only approximate as different effects physically do not follow a

<sup>10</sup> The ‘Telescope’ nomenclature is introduced to separate the AOCS model and the optical effects in the software architecture.

## A PlatoSim simulation: from Initialisation to Computation



**Fig. 15.** Schematic of the PlatoSim software package. (a) Overview of initialisation and configuration of PlatoSim prior to simulation execution. (b) Overview of each simulation step as a loop over the total number of exposures. The boxes represent input files (purple), the output file (orange), software modules (blue), and the general simulation steps (green). The two flowcharts combined illustrates PlatoSim’s events of execution by (a) first constructing a simulation (and all input parameters needed) followed by (b) creating synthetic pixel images for a given number of exposures.

deterministic entry point of occurrence which algorithmic architectures necessarily need to comply to. For example various CCD and FEE effects (such as BFE, blooming, and non-linearity) strongly depend on each other. As a final action of each acquisition, the photometry module is (optionally) applied, the output to written to disk, and the internal clock is updated.

Lastly, PlatoSim allows for the possibility of parallelisation. The execution of a simulation on multiple cores (or CPUs) is not only possible per subfield (as was done for the simulations in Sect. 11), but also in time. The former execution is a standard sequential workflow (i.e. where each simulation run independently on a designated core), whereas the latter options is a partitioning (i.e. a single simulation is chopped into smaller time series and deliver each to a different core). The latter option is more complex as it requires the time series of any supplementary input file (e.g. the variable source file) to be computed ahead of the simulation, however, the random seeds are handled automatically by PlatoSim during the partitioning.

### 11. Applications to the PLATO mission

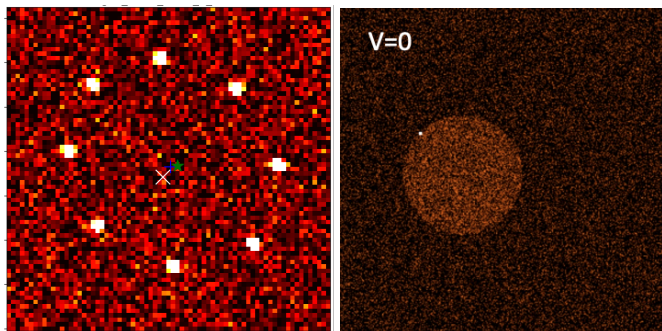
Due to the end-to-end and modular design, PlatoSim is used for many different disciplines within the PLATO mission consortium as will be discussed in the following. We note that the

data and results of Sects. 11.3 and 11.4 are made available to everyone through the PlatoSim repository.

#### 11.1. Mission preparation

The assembly, integration, verification and test (AIV/AIT) of a spacecraft such as PLATO involves a considerable amount of preparatory work. This is done first of all to design the details of operations for each of the tests, which will impact the quality and the relevance of the results as well as the planning. Secondly, preparatory work also helps defining the necessary Ground Support Equipment (optical stimuli, telemetry acquisition systems, etc.), designing the data processing algorithm for each test, etc. PlatoSim played a key role for a series of tests in this respect.

The first delicate operation when assembling PLATO is to align the cameras. This involves assembling the FPA, bearing the detectors with the TOU within very tight accuracy budgets (Pertenais et al. 2021), and ensuring the optimal performance of the camera in operations. While the PLATO cameras will operate in vacuum and around  $-70^{\circ}\text{C}$ , the alignment and assembly occur at room temperature, that is  $\sim 100^{\circ}\text{C}$  warmer and under normal atmospheric pressure. Consequently, during the alignment, not only is the optical quality of the cameras considerably degraded, but the detectors also produce tens of



**Fig. 16.** Examples of simulated data generated in preparation of the AIT/AIV of PLATO mission. Left: Simulation of a Hartmann pattern obtained at ambient temperature and far out of focus, in preparation of the camera alignment. Right: Simulation of a  $V = 0$  star close to the optical axis of the camera (white dot) and the extended ghost it creates on the detector via parasitic reflections, ran in preparation of the thermal-vacuum tests.

thousands of  $e^- s^{-1} \text{pixel}^{-1}$  of dark current. The optical verification necessary for a proper alignment hence requires dedicated operational modes. This includes short integration times and partial readout of the CCDs to limit the dark current, CCD clearout between every frame, continuous image-dump between observations, etc. PlatoSim was used to simulate data obtained at ambient temperature to make the trade-off between observations under full pupil illumination and observations with a hartmann mask. Hence, PlatoSim was used to design the test approach and data-reduction process, and estimate the resulting accuracy in each case. The details on the simulations and on the alignment of the engineering camera are presented in Royer et al. (2020) and Royer et al. (2022), respectively.

PlatoSim is also instrumental in the preparation of the environmental tests performed at thermal vacuum (TVAC). We here cite four examples where PlatoSim was used in the testing. First the long term stability test, aimed at testing the measurement stability at camera-level. Secondly, the characterisation of the image-ghosts due to multiple reflections in between various optical surfaces and on the detector (Pertenais et al. 2022). Third, the characterisation of the camera image geometry, that is of the optical distortions induced by the very wide field. Finally and fourth, the optimisation of the measurement strategy for the critical but time consuming determination of the best focus temperature, and characterisation of the image quality at the optimal focus temperature (Borsa et al. 2022).

PlatoSim is also used to prepare the operations at spacecraft level, for instance in simulations to test the compression algorithms to run on the Instrument Control Unit on-board the spacecraft, or in simulations of real-time operations of the Fine Guiding System, that is in the feedback loop between the fast cameras and the AOCS (Grießbach et al. 2021). Figure 16 displays an example of a hartmann pattern simulated for an ambient temperature test (left), as well as a simulation of the extended ghost image caused by an very bright star (right).

### 11.2. Pipeline validation

According to PLATO’s main objectives the pipeline chain needs to be able to preserve signals in the light curve belonging to transiting planets, stellar activity, and stellar pulsations. These are time varying phenomena ranging from seconds to months. PLATO must deal with a huge photometric dynamical range of

more than ten orders of magnitude. Furthermore, a strategy for merging the light curves across multiple cameras and mission quarters, in combination with the limited telemetry capabilities, require that most of the scientific analysis must be done autonomously on-board the spacecraft. Altogether this demands a sublime pipeline performance. The construction of a versatile pipeline can only be done prior to launch using a test harness of simulations for which PlatoSim plays a key role.

In comparison to the built in pre-processing steps of PlatoSim’s on-board photometry module (see Sect. 9.2), the removal of outliers and flux corrections from short term and long term pixel displacements are algorithms yet to be implemented. We highlight that the full PLATO pipeline chain consist of three main branches for the light curve generation: i) on-ground; ii) on-board; and iii) for saturated stars (Rauer et al. 2014, and in prep.). To accommodate the need for generating fully calibrated data product of non-saturated stars, a computationally bridge between PlatoSim and the (preliminary) PLATO reduction and extraction pipeline has been established.

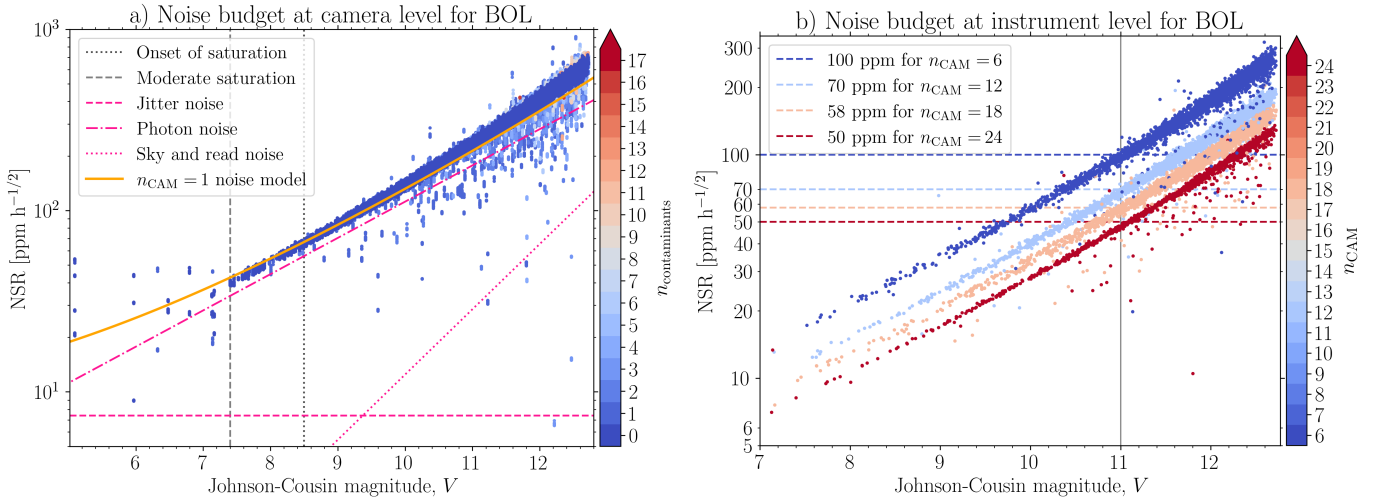
For the validation of the photometric extraction of saturated stars PlatoSim is heavily used due to its ability to simulate non-linear effects. To validate the pipeline performance of non-saturated stars, stitching and detrending of light curves is required. Indeed, a wealth of events will leave large data gaps and introduce systematic errors that can be highly correlated and thus almost impossible to model and remove. Poor attempts can easily hinder the detection of the astrophysical signals. Hence light curve stitching and detrending are well studied topics in space photometry (García et al. 2011; Vanderburg & Johnson 2014; Handberg & Lund 2014; Lund et al. 2015, 2021; Aigrain et al. 2017; Hippke et al. 2019). They are also extremely instrument dependent and science driven and thus deserve special attention for PLATO. As such PlatoSim simulations are currently being used for this aspect of the pipeline, which is vital for PLATO’s ability to characterise the exoplanet host stars, and thus ultimately the planets themselves.

### 11.3. Performance studies

The capacity to investigate how the photometric precision internally depends on instrumental systematics on both short and long time scales shorter than the cadence is one of PlatoSim’s advantages. As an example we show a performance study of PLATO’s expected photometric precision at BOL.

With a premise that a 24 h duration dataset is representative for estimating the averaged NSR of each light curve, we simulated 10 000 F5-K7 dwarf and sub-dwarf stars from the LOP south covering a large photometric dynamical range. To populate the NSR– $V$  diagram approximately evenly a total number 2 500 stars was drawn from their camera observability  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$ , meaning that in total 150 000 light curves were generated. To simulate the effect of stellar crowding a choice was made to include photometric contaminant stars brighter than  $\Delta V < 5$  and within a relative radial distance to their target of 45 arcsec, that is maximally three pixels away from the target barycentre within each simulated imagette.

All simulations are configured with a realistic AOCS jitter time series from OHB/TAS sampled at 8 Hz (being the model shown in the bottom panel of Fig. 8) to realistically include its impact on the photometric precision ‘as expected’. All random and systematic noise sources are configured ‘as required’ of the instrument design. For computational alleviation the analytic PSF model was used and charge diffusion was activated. Furthermore, it is assumed that the properties of each TOU, CCD,



**Fig. 17.** NSR( $V$ ) simulation study at BOL as required by the mission. (a) Noise budget at the camera level with each data point coloured after the number of stellar contaminants contained within 3 pixel and  $\Delta V < 5$  of the target star. The model prediction of the noise (orange solid curve) consists of three photometric noise components: jitter noise (pink dashed line calculated using an averaged (yaw, pitch, and roll) jitter time series rms amplitude value of 0.037 arcsec) dominating in the bright regime, readout and sky background noise (pink dotted line using a sky background value of around  $60 \text{ e}^- \text{ pixel}^{-1} \text{ s}^{-1}$ ) dominating in the faint regime, and photon noise (pink dashed-dotted line) dominated between the two aforementioned regimes. Also the onset of saturation is indicated (grey dotted line) together with the onset of moderate saturation (grey dotted line) here defined as where an extended mask expanding beyond the dimensions of an imagette is needed to capture total stellar flux due to blooming. (b) Noise budget at the instrument level with each data point representing a multi-camera observation colour by the number of N-CAM observations used in the NSR calculation. The mission requirement of the photometry for  $V < 11$  is shown for an observability of  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$  (horizontal dashed lines coloured after  $n_{\text{CAM}}$ ).

and FFE among all cameras are identical and their noise sources are uncorrelated. However, we randomly displace each camera pointing to imitate camera misalignments to the interface of the optical bench. For all simulations a constant intrinsic intensity was considered and the photometric extraction was performed by the build-in photometry module of *PlatoSim* (as described in Sect. 9) due to its superior computational speed compared to the preliminary on-board *PLATO* pipeline.

For each star we compute the NSR by first combining the normalised light curves from each camera, averaging measurements of the same camera group (as they have identical time stamps). Next we resample the data into 1 h bins, and compute the standard deviation. The resulting photometric noise is shown in Fig. 17a for each individual camera observation (i.e. at camera level) and Fig. 17b for the multi-camera observations (i.e. at instrument level).

Taking a closer look at Fig. 17a the colours represent the total number of stellar contaminants included within each simulation and the orange line is NSR model prediction given the mission requirements at BOL. In detail this model contain a component of noise from AOCS jitter (pink dashed line), photon noise (pink dashed-dotted line), and sky plus read noise (pink dotted line). As expected for dynamic range simulated, the photon noise dominated the noise budget, however, random noise from the sky background and readout noise will ultimately dominate beyond  $V > 15$ . The simulations shows a slight discrepancy with the model prediction in the bright end where noise from AOCS jitter starts to dominate. A thorough investigation shows a clear indication that the NSR at the onset of pixel saturation (grey dotted line) only starts to dependent on the CCD properties, such as the barycentric location of star, for relatively high rms amplitudes of the jitter Euler angles. The photometric prediction for saturated stars may thus behave very different to the in-flight measurements (especially considering asymmetric blooming not included here), nevertheless, it is clear that the

photometric precision extracted from imagettes depends on the combination of saturation and AOCS jitter with an ultimate noise floor set for  $V < 7.4$ . This relates to the onset of moderate saturation (grey dotted line), as defined here, to when blooming cause flux leakage out of an imagette and an extended mask to conserve the flux measurement (for which an extended mask library are currently being designed using *PlatoSim*).

For the NSR at instrument level, Fig. 17b displays a clear division between stars observed with  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$ . The applied camera misalignment model introduces significant barycentric displacements up to a few tens of pixels from camera to camera, which in turn may render some stars unobservable with one or more cameras (as expected in-flight). The figure illustrates that the mean noise level for stars with  $V < 11$  satisfy the requirements of  $\text{NSR} \leq \{100, 70, 58, 50\} \text{ ppm h}^{-1/2}$  for observations with  $n_{\text{CAM}} \in \{6, 12, 18, 24\}$ , respectively. This agrees with a similar simulation study of Rauer et al. (in prep.) using the *PINE* simulator (Börner et al. 2022).

*PINE* is a theoretical noise estimator that uses the true (i.e. uncontaminated) flux of the target star to calculate the NSR, meanwhile such an approach is not readily possible at pixel level whilst extracting the photometry of stars from real stellar fields (such as that of the *PIC*). Since flux leakage from stellar contaminant(s) into the aperture mask is unavoidable (and can only partially be mitigated by the mask definition itself), the scattered data points below each of the corresponding NSR- $n_{\text{CAM}}$  curves of Fig. 17b are *PIC* targets with one or more contaminants (as shown in Fig. 17a). Naturally, accounting for stellar contamination is expected to place each of these measurements, above the corresponding NSR curve due to the addition of photon noise from the stellar contaminant(s).

#### 11.4. Hare and Hound exercises

Hare and Hound exercises, also known as injection and retrieval exercises, have more pragmatically been performed with other

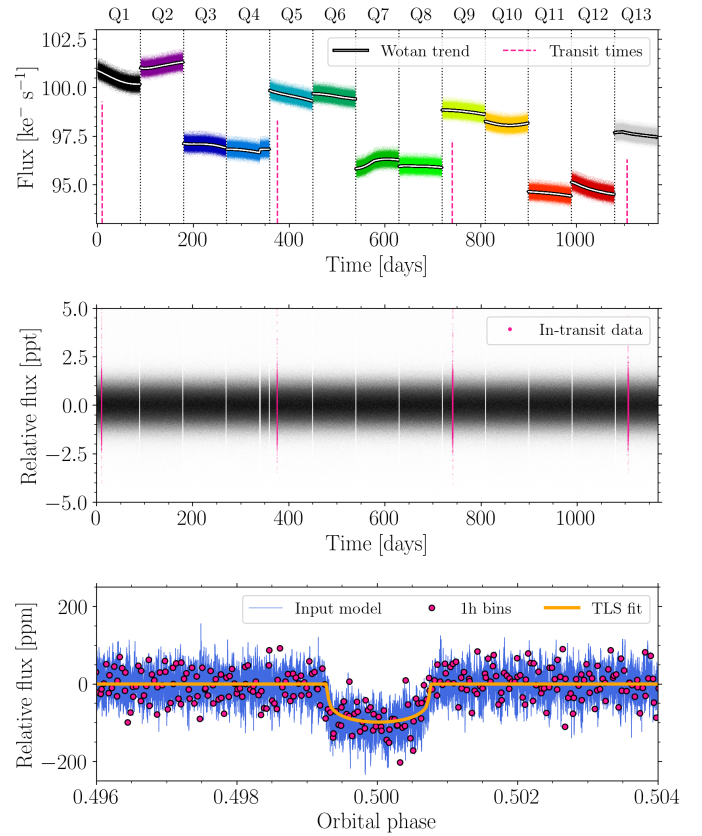
PLATO simulators (such as PSLs; Samadi et al. 2019). Although a full suite of such exercises using PLATOsim is beyond the scope of this paper, we make a realistic show case of PLATO’s ability to detect an Earth-like planet orbiting a Sun-like host star. We simulate a  $V = 10$  star (i.e.  $\mathcal{P} \approx 10.4$  cf. Eq. (13)) observed with all 24 N-CAMs. We simulate 13 mission quarters ( $\sim 3.3$  yr) including four transits to test the retrieval efficiency of 2, 3, and 4 transits.

We employ the same computational setup as in Sect. 11.3 with a change of the AOCS jitter to a red noise model sampled at 0.1 Hz, and an activation of the KDA model. The TED for each camera and quarter is included using a second order polynomial model whilst uniformly drawing the model coefficients under the restriction that the amplitude in yaw, pitch, and roll cannot exceed 10 arcsec. The latter is a conservative choice and was deliberately made to challenge the software that corrects for systematic long-term trends in the light curves. For a realistic representation of the noise budget for our P1 sample star, the preliminary PLATO pipeline (using PSF photometry) is employed.

As the majority of targets from the PIC are expected to have convection-driven variability we model and inject stellar granulation and oscillations into the simulated target. A full asteroseismic Hare and Hound exercise is out of the scope of this paper, but a full model description of the solar-like oscillator is presented in Appendix C. For the present showcase only the impact of the noise floor of stellar variability on the transit detection is of interest.

To generate the planet transits we use the open source software batman<sup>11</sup> (Kreidberg 2015) tuning the radius, mass, and orbital period ( $P = 365.25$  d) to that of the Sun-Earth system (assuming no third bodies) and choose the time of ephemeris to  $t_0 = 10$  d. For simplicity we consider circular orbits ( $e = 0$ ) and edge-on transits ( $i = 90^\circ$ ). We use a quadratic limb darkening (LD) law and calculate the LD coefficients using the software PyLDtk<sup>12</sup> (Parviainen & Aigrain 2015) to calculate the stellar LD profile in the PLATO passband. For our simulated Earth analogue orbiting a G2V host star this results in a transit depth of  $\delta \approx 103$  ppm and a total transit duration of  $T_{\text{tot}} \approx 13$  h. The transit depth overshoot of  $\sim 22\%$  compared to the planet-to-star radius ratio of 84 ppm has been explained by the effect of stellar limb darkening (Heller 2019).

We follow the transit retrieval procedure of Heller et al. (2022) and use the open source software Wotan<sup>13</sup> (Hippke et al. 2019) for detrending. Wotan is optimised to preserve transiting signatures of exoplanets while effectively removing instrumental and stellar variability using a large library of available detrending filters. We here use Turkey’s biweight method and a window size of  $3 \times T_{\text{tot}}$  shown by Hippke et al. (2019) to be an optimal choice for most transit searches. The detrending is performed on each individual camera and mission quarter segment. As a performance illustration of the detrending the top panel of Fig. 18 shows the simulated time series for a single camera across 13 mission quarters (coloured segments), the corresponding model trend (white/black line), and the mid transit times (dashed pink lines). Aside from the model trends caused primarily by the TED, Fig. 18 shows two additional dominating features; (i) large jumps in the mean flux level from one mission quarter to the next caused by a change in the optical throughput as the quarterly rotations relocate the stars to different distances



**Fig. 18.** Results of the Hare and Hound (injection and retrieval) exercise of an Earth-sized planet transiting a  $V = 10$  Sun-like star. With an orbital period of 365.25 d the planet transits 4 times in the 3.3 yr simulated light curve. Top: Light curves observed with a single camera (coloured per quarter) and the corresponding Wotan trend (white/black lines). Middle: detrended light of all 24 camera observations (black dots) with in-transit data highlighted (dots). Bottom: Phase folded light curve on the best fit period from TLS. For clarity a 1 h binned representation of the light curve is shown (pink circles), together with the injected variability (blue line), and a best fit transit model (orange line).

from the optical axis in the FPA (whilst the corresponding change in PRNU is a second order effect here); and (ii) an overall decrease in intensity due to the combined set of ageing effects.

The final detrended light curve merged across all cameras and quarters is shown in the middle panel of Fig. 18 and is used for the transit vetting. The actual transit search is performed using the transit least-squares (TLS) method implemented in the open software TLS<sup>14</sup> (Hippke & Heller 2019). Compared to the traditional box least-squares (BLS) method, TLS outperforms the BLS algorithm (Kovács et al. 2002) especially in the domain of small planets (e.g. Heller et al. 2019b,a) as it includes a precise modelling of the transiting signature (in particular the ingress, egress, and the LD profile). Upon execution TLS performs a grid search in the parameter space  $\{P, t_0, T_{\text{tot}}\}$  to find the minimum  $\chi^2$  value. While doing so, it calculates the Signal Detection Efficiency (SDE) being the key search metric of how significant the  $\chi^2$  minimum is compared to its surrounding  $\chi^2$  landscape as a function of orbital period.

Before employing the actual vetting a  $4\sigma$  clipping filter was used to remove large outliers and all measurements from the same camera group were averaged. Lastly, a 1 h binned representation of the retrieved light curve was created (which here

<sup>11</sup> <https://github.com/lkreidberg/batman>

<sup>12</sup> <https://github.com/hpparvi/ldtk>

<sup>13</sup> <https://github.com/hippke/wotan>

<sup>14</sup> <https://github.com/hippke/tls>

is justified for the planet detection however not for characterisation). The TLS vetting of 2, 3, and 4 transits resulted respectively in a SDE of around 24, 40, and 47. These are well beyond the previous reported detection threshold of  $SDE \geq 9$  that result in a false-alarm probability of  $FPR < 10^{-4}$  for Earth-sized planets (Hippke & Heller 2019). As an illustration the bottom panel of Fig. 18 shows the phase-folded light curve of the best TLS fit to the full dataset (orange line) together with the injected variability model (blue line) and a the 1 h binned light curve (pink dots). With a signal-to-noise (S/N) ratio of  $S/N > 7$  (cf. Pont et al. 2006) in all above cases the presence of stellar granulation and pulsations does not avert the planet retrieval, which agrees well with the simulation study of Morris et al. (2020).

## 12. Discussion and conclusion

PlatoSim is an advanced end-to-end CCD and light-curve simulator dedicated to simulate PLATO measurements. Its algorithms and their underlying methodology aim to model realistic space-based photometric imaging. PlatoSim is available on GitHub<sup>15</sup> together with a detailed documentation and tutorials.

The mission requirements of PLATO's innovative multi-camera configuration pushes the frontier of the instrumental design. Reliable simulations are therefore essential – from the early design phases all the way to the in-flight operations – to continuously assist with the design, assessment, and validation of the instrument. PlatoSim was developed to accommodate for this niche exactly. We have provided several examples of applications, ranging from dedicated studies to address the technical assessment and verification of the instrument (Sect. 11.1), the development of data reduction and processing algorithms (Sect. 11.2), the performance assessment of the payload (Sect. 11.3), to the verification of a core science case relevant to the mission (Sect. 11.4).

To our knowledge, PlatoSim is one of the most feature-rich simulators currently available to simulate space-based photometry, including a wide range of instrumental noise sources both at platform level (e.g. AOCS jitter and TED), camera level (e.g. optical distortion and ghost images), and detector level (e.g. PRNU, CTI, and BFE), as well as astrophysical signals (granulation, stochastic oscillations, and exoplanet transits). A unique ability is the inclusion of a realistic PSF (as illustrated in Fig. 15b) either from Zemax simulations, or from a parametric description allowing a PSF that varies over the focal plane, while keeping the computational burden feasible. PlatoSim stands out due to its configurability of the N-CAM and F-CAM to simulate: (i) a full suite of PLATO data products (i.e. images, meta data, housekeeping data, and light curves); (ii) the different modes of operation (e.g. nominal observations and microscanning sessions); and (iii) the different payload configurations. This together with its end-to-end functionality to generate light curves coming from 24 cameras makes it a key simulator for the PLATO mission consortium.

PlatoSim chooses to perform its simulations in the time domain rather than in the Fourier domain. The advantage is the ability to easily include time-dependent instrumental variations that may interfere with the stellar signal. The corresponding challenge is the computational resources needed to generate long time series. PlatoSim largely manages to overcome the computational bottlenecks and drastically reduce the execution time by making extensive use of efficient parametric models

while still preserving a realistic simulation of each physical phenomenon. An additional mitigation effort is the coherent use of wavelength-averaged quantities such as the PSF, the throughput, the detector efficiency, and the quantum efficiency, to avoid a time-consuming numerical integration over the wavelength in the PLATO passband. The monochromatic light approximation, however, makes it difficult to compare the increased level of noise expected for the extreme spectral types of PLATO's targets (i.e. F5 and K7) had the wavelength dependence of the above-mentioned quantities been taken into account. Nevertheless, as PlatoSim is designed to simulate stars of spectral types similar to the Sun, quantitatively the impact on the photometry is expected to be small. Given PLATO's spectral response, the baseline at camera level will be that the noise in the light curve of a F5 dwarf star will be slightly higher than for a K7 dwarf star.

Despite already being an advanced simulator, PlatoSim remains a simulation tool that is continuously being improved and extended upon. At the time of writing, the AIV of the PLATO cameras is well under way, and laboratory measurements of the instrument characteristics such as the PRNU and the morphology of ghost images are becoming available. These measured quantities will in the near future replace any simulated quantities in PlatoSim where possible. On the topic of the PRNU, if inter-pixel sensitivity measurements become available, they will allow PlatoSim to accurately account for a lower pixel sensitivity near the pixel borders opposed to the pixel centre. With PlatoSim's ability to track time-dependent effects on time scales shorter than the exposure time, such an improved PRNU model will play a key role for future development concerning effects that are best described at the subpixel level.

Charge injection will be implemented in order to investigate its potential as mitigation of the CTI in the final stage of the mission's lifespan when the CTI could be significantly worse depending on the solar activity. Additionally, a relevant noise source for photometric measurements is scattered light coming from the Earth and the Moon. Although measures will be taken to avoid this as much as possible, some residual light scatter may still occur. Since this scatter is time dependent and could interfere with the detected stellar signal, it is worth considering and implementing this in PlatoSim. Defective pixels are also on the list of future implementations. In particular, RTS pixels are important as their variable dark current behaviour for slow state switching (hours to days) will introduce flux jumps (similar to mask updates) and for fast state switching (from seconds to minutes) can introduce a significantly increased scatter in the extracted light curves. Finally, PlatoSim advantageously allows the user to simulate CCD subfields instead of full-frame images to reduce computational resources. As such, smearing and blooming trails from stars outside the subfield, plus light contribution from stars close to the pixel borders of the subfield, are not accounted for. Since the former is particularly important for designing the processing algorithms that account for the impact of smearing, it is a feature that belongs among the future enhancements of PlatoSim.

With the PLATO mission planned to see first light in less than 3 yr the scientific justifications for the core and complementary science programmes are constantly being verified with simulations. As demonstrated in this work, PlatoSim offers the opportunity to validate the performance of the PLATO payload and to undertake detailed simulation studies with a highly realistic approach that includes all foreseen random and systematic noise sources, an exact distribution of stars across the multi-camera arrangement in line with future planned pointing fields, and the inclusion of stellar variability for both target

<sup>15</sup> <https://github.com/IvS-KULeuven/PlatoSim3/tree/master> (for PLATO mission consortium members).

and contaminant stars. This makes PlatoSim a versatile software package under continuous development as a bedrock in the preparation for PLATO's future discoveries.

**Acknowledgements.** This work presents results from the European Space Agency (ESA) space mission PLATO. The PLATO payload, the PLATO Ground Segment and PLATO data processing are joint developments of ESA and the PLATO mission consortium (PMC). Funding for the PMC is provided at national levels, in particular by countries participating in the PLATO Multilateral Agreement (Austria, Belgium, Czech Republic, Denmark, France, Germany, Italy, Netherlands, Portugal, Spain, Sweden, Switzerland, Norway, and United Kingdom) and institutions from Brazil. Members of the PLATO Consortium can be found at <https://platomission.com/>. The ESA PLATO mission website is <https://www.cosmos.esa.int/plato>. We thank the teams working for PLATO for all their work. The research behind these results has received funding from the Belgian federal Science Policy Office (BELSPO) through PRODEX grant PLATO: ZKE2050-01-D01. R.H. acknowledges support from the German Aerospace Agency (Deutsches Zentrum für Luft- und Raumfahrt) under PLATO Data Center grant 50001501. This project additionally made use of the following published Python packages: NumPy (Harris et al. 2020), Numba (Lam et al. 2015), Pandas (McKinney et al. 2011; The pandas development team 2022), SciPy (Virtanen et al. 2020), Matplotlib (Hunter 2007), and Astropy (Astropy Collaboration 2022, 2018, 2013).

## References

- Aerts, C. 2021, *Rev. Mod. Phys.*, **93**, 015001
- Aerts, C., Christensen-Dalsgaard, J., & Kurtz, D. W. 2010, *Asteroseismology* (Springer Science and Business Media)
- Aigrain, S., Parviainen, H., & Pope, B. J. S. 2016, *MNRAS*, **459**, 2408
- Aigrain, S., Parviainen, H., Roberts, S., Reece, S., & Evans, T. 2017, *MNRAS*, **471**, 759
- Allison, J., Amako, K., Apostolakis, J., et al. 2016, *Nucl. Instrum. Methods Phys. Res. A*, **835**, 186
- Anderson, J., & King, I. R. 2006, *PSFs, Photometry, and Astronomy for the ACS/WFC*, Instrument Science Report ACS 2006-01, 34
- Anderson, E. R., Duvall, Thomas L., J., & Jefferies, S. M. 1990, *ApJ*, **364**, 699
- Antilogus, P., Astier, P., Doherty, P., Guyonnet, A., & Regnault, N. 2014, *J. Instrum.*, **9**, C03048
- Arentoft, T., Kjeldsen, H., De Ridder, J., & Stello, D. 2004, in *ESA Spec. Publ.*, 538, *Stellar Structure and Habitable Planet Finding*, eds. F. Favata, S. Aigrain, & A. Wilson, 59
- Arko, M., Prod'homme, T., Lemmel, F., et al. 2022, *J. Astron. Telescopes Instrum. Syst.*, **8**, 048002
- Astier, P., & Regnault, N. 2023, *A&A*, **670**, A118
- Astier, P., Antilogus, P., Juramy, C., et al. 2019, *A&A*, **629**, A36
- Astropy Collaboration (Robitaille, T. P., et al.) 2013, *A&A*, **558**, A33
- Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, *AJ*, **156**, 123
- Astropy Collaboration (Price-Whelan, A. M., et al.) 2022, *AJ*, **935**, 167
- Auvergne, M., Bodin, P., Boissard, L., et al. 2009, *A&A*, **506**, 411
- Bély, P., Hasan, H., Miebach, Manfred Bély, P., Hasan, H., & Miebach, M. 1993, *Orbital Focus Variations in the Hubble Space Telescope*, Instrument Science Report SESD 93-16, 9
- Benz, W., Broeg, C., Fortier, A., et al. 2021, *Exp. Astron.*, **51**, 109
- Blanco Rodríguez, J., del Toro Iniesta, J. C., Orozco Suárez, D., et al. 2018, *ApJS*, **237**, 35
- Börner, A., Paproth, C., Cabrera Perez, J., et al. 2022, *Research Square*, <https://doi.org/10.21203/rs.3.rs-2159430/v1>
- Borsa, F., Cottinelli, A., Gorius, N., et al. 2022, *SPIE Conf. Ser.*, **12180**, 121801D
- Borucki, W. J., Koch, D., & Kepler Science Team 2010, *Am. Astron. Soc.*, **42**, 47.03
- Bowman, D. M. 2020, *Front. Astron. Space Sci.*, **7**, 70
- Bradley, L., Sipocz, B., Robitaille, T., et al. 2016, *Astrophysics Source Code Library*, [record ascl:1609.011]
- Brown, D. C. 1971, *Eng. Rem. Sens.*, **37**, 855
- Brown, T. M., & Gilliland, R. L. 1994, *ARA&A*, **32**, 37
- Brun, A. S., García, R., Houdek, G., Nandy, D., & Pinsonneault, M. 2015, *Space Sci. Rev.*, **196**, 303
- Bryson, S. T., Tenenbaum, P., Jenkins, J. M., et al. 2010, *ApJ*, **713**, L97
- Bryson, S., Kunitomo, M., Kopparapu, R. K., et al. 2021, *AJ*, **161**, 36
- Campante, T. L., Schofield, M., Kuszewicz, J. S., et al. 2016, *ApJ*, **830**, 138
- Chaplin, W. J., & Basu, S. 2014, *Space Sci. Rev.*, **186**, 437
- Chaplin, W. J., Appourchaux, T., Elsworth, Y., et al. 2010, *ApJ*, **713**, L169
- Chaplin, W. J., Kjeldsen, H., Christensen-Dalsgaard, J., et al. 2011, *Science*, **332**, 213
- Christensen-Dalsgaard, J., Kjeldsen, H., Brown, T. M., et al. 2010, *ApJ*, **713**, L164
- Coelho, P., Barbuy, B., Meléndez, J., Schiavon, R. P., & Castilho, B. V. 2005, *A&A*, **443**, 735
- Corsaro, E., Fröhlich, H. E., Bonanno, A., et al. 2013, *MNRAS*, **430**, 2313
- Cunha, M. S., Aerts, C., Christensen-Dalsgaard, J., et al. 2007, *A&ARv*, **14**, 217
- Da Costa, G. S. 1992, in *ASP Conf. Ser.*, **23**, *Astronomical CCD Observing and Reduction Techniques*, ed. S. B. Howell, 90
- Dauser, T., Falkner, S., Lorenz, M., et al. 2019, *A&A*, **630**, A66
- De Ridder, J., Arentoft, T., & Kjeldsen, H. 2006, *MNRAS*, **365**, 595
- Drummond, R., Vandenbusche, B., Aerts, C., De Oliveira Fialho, F., & Auvergne, M. 2006, *PASP*, **118**, 874
- Fairfield, J., Groom, D., Bailey, S., et al. 2007, *Nucl. Sci. IEEE Trans.*, **53**, 3877
- Futyan, D., Fortier, A., Beck, M., et al. 2020, *A&A*, **635**, A23
- García, R. A., & Ballot, J. 2019, *Liv. Rev. Sol. Phys.*, **16**, 4
- García, R. A., Mathur, S., Salabert, D., et al. 2010, *Science*, **329**, 1032
- García, R. A., Hekker, S., Stello, D., et al. 2011, *MNRAS*, **414**, L6
- Gilliland, R. L., Brown, T. M., Christensen-Dalsgaard, J., et al. 2010, *PASP*, **122**, 131
- Grießbach, D., Witteck, U., & Paproth, C. 2021, *SPIE Conf. Ser.*, **11852**, 118523H
- Guyonnet, A., Astier, P., Antilogus, P., Regnault, N., & Doherty, P. 2015, *A&A*, **575**, A41
- Handberg, R., & Lund, M. N. 2014, *MNRAS*, **445**, 2698
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Nature*, **585**, 357
- Hedges, C., Luger, R., Martinez-Palomera, J., Dotson, J., & Barentsen, G. 2021, *AJ*, **162**, 107
- Hekker, S. 2013, *Adv. Space Res.*, **52**, 1581
- Heller, R. 2019, *A&A*, **623**, A137
- Heller, R., Hippke, M., & Rodenbeck, K. 2019a, *A&A*, **627**, A66
- Heller, R., Rodenbeck, K., & Hippke, M. 2019b, *A&A*, **625**, A31
- Heller, R., Harre, J.-V., & Samadi, R. 2022, *A&A*, **665**, A11
- Hill, M. L., Bott, K., Dalba, P. A., et al. 2023, *AJ*, **165**, 34
- Hippke, M., & Heller, R. 2019, *A&A*, **623**, A39
- Hippke, M., David, T. J., Mulders, G. D., & Heller, R. 2019, *AJ*, **158**, 143
- Hon, M., Huber, D., Kuszewicz, J. S., et al. 2021, *ApJ*, **919**, 131
- Howell, S. B. 1989, *PASP*, **101**, 616
- Howell, S. B., Sobek, C., Haas, M., et al. 2014, *PASP*, **126**, 398
- Hoyer, S., Guterman, P., Demangeon, O., et al. 2020, *A&A*, **635**, A24
- Huber, D., Chaplin, W. J., Christensen-Dalsgaard, J., et al. 2013, *ApJ*, **767**, 127
- Hunter, J. D. 2007, *Comput. Sci. Eng.*, **9**, 90
- Husser, T. O., Wende-von Berg, S., Dreizler, S., et al. 2013, *A&A*, **553**, A6
- Israel, H., Massey, R., Prod'homme, T., et al. 2015, *MNRAS*, **453**, 561
- Janesick, J. R. 2001, *Scientific charge-coupled devices* (Bellingham, WA: SPIE Optical Engineering Press)
- Jorden, P. R., Deltorn, J. M., & Oates, A. P. 1994, *SPIE Conf. Ser.*, **2198**, 836
- Kallinger, T., De Ridder, J., Hekker, S., et al. 2014, *A&A*, **570**, A41
- Kallinger, T., Hekker, S., García, R. A., Huber, D., & Matthews, J. M. 2016, *Sci. Adv.*, **2**, 1500654
- Kiefer, R. 2022, *44th COSPAR Scientific Assembly*, 16–24 July, **44**, 2488
- Kjeldsen, H., & Bedding, T. R. 1995, *A&A*, **293**, 87
- Kjeldsen, H., & Frandsen, S. 1992, *PASP*, **104**, 413
- Klaassen, P. D., Geers, V. C., Beard, S. M., et al. 2021, *MNRAS*, **500**, 2813
- Kovács, G., Zucker, S., & Mazeh, T. 2002, *A&A*, **391**, 369
- Kreidberg, L. 2015, *PASP*, **127**, 1161
- Lam, S. K., Pitrou, A., & Seibert, S. 2015, in *Proc. Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1
- Lanman, A., Hazelton, B., Jacobs, D., et al. 2019, *J. Open Source Softw.*, **4**, 1234
- Lawrence, D., O'Connor, P., Frank, J., & Takacs, P. 2011, *PASP*, **123**, 1100
- Leinert, C., Bowyer, S., Haikala, L. K., et al. 1998, *A&AS*, **127**, 1
- Leschinski, K., Czoske, O., Köhler, R., et al. 2016, *SPIE Conf. Ser.*, **9911**, 991124
- Libralato, M., Bedin, L. R., Nardiello, D., & Piotto, G. 2016, *MNRAS*, **456**, 1137
- Lightkurve Collaboration, Cardoso, J. V. d. M., Hedges, C., et al. 2018, *Astrophysics Source Code Library*, [record ascl:1812.013]
- Lorente, N. P. F., Glasse, A. C. H., Wright, G. S., & García-Marín, M. 2006, *SPIE Conf. Ser.*, **6274**, 62741F
- Lucsanyi, D., & Prod'homme, T. 2020, *IEEE Trans. Nuclear Sci.*, **67**, 1623
- Lund, M. N., Handberg, R., Davies, G. R., Chaplin, W. J., & Jones, C. D. 2015, *ApJ*, **806**, 30
- Lund, M. N., Handberg, R., Buzasi, D. L., et al. 2021, *ApJS*, **257**, 53
- Marchiori, V., Samadi, R., Fialho, F., et al. 2019, *A&A*, **627**, A71
- Marcos-Arenal, P., Zima, W., De Ridder, J., Huygen, R., & Aerts, C. 2014a, *ArXiv e-prints* [arXiv:1402.2582]
- Marcos-Arenal, P., Zima, W., De Ridder, J., et al. 2014b, *A&A*, **566**, A92
- Massey, R., Schrabback, T., Cordes, O., et al. 2014, *MNRAS*, **439**, 887
- McKinney, W. et al. 2011, *Python for High Performance and Scientific Computing*, **14**, 1
- Michel, E., Baglin, A., Weiss, W. W., et al. 2008, *Commun. Asteroseismol.*, **157**, 69
- Montalto, M., Piotto, G., Marrese, P. M., et al. 2021, *A&A*, **653**, A98



- Morris, B. M., Bobra, M. G., Agol, E., Lee, Y. J., & Hawley, S. L. 2020, *MNRAS*, **493**, 5489
- Nascimbeni, V., Piotto, G., Börner, A., et al. 2022, *A&A*, **658**, A31
- Naylor, T. 1998, *MNRAS*, **296**, 339
- O'Brien, A., Beard, S., Geers, V., & Klaassen, P. 2016, *SPIE Conf. Ser.*, **9913**, 991312
- Park, S. C., Park, M. K., & Kang, M. G. 2003, *IEEE Signal Process. Mag.*, **20**, 21
- Parviainen, H., & Aigrain, S. 2015, *MNRAS*, **453**, 3821
- Pertenais, M., Cabrera, J., Paproth, C., et al. 2021, *SPIE Conf. Ser.*, **11852**, 118524Y
- Pertenais, M., Ammler-von Eiff, M., Burresi, M., et al. 2022, *SPIE Conf. Ser.*, **12180**, 121804M
- Pont, F., Zucker, S., & Queloz, D. 2006, *MNRAS*, **373**, 231
- Popowicz, A. 2018, *SPIE Conf. Ser.*, **10698**, 1069820
- Prod'homme, T., Verhove, P., Beaufort, T., et al. 2016, *SPIE Conf. Ser.*, **9915**, 99150U
- Prod'homme, T., Verhove, P., Lemmel, F., et al. 2018, *SPIE Conf. Ser.*, **10709**, 107090K
- Rauer, H., Catala, C., Aerts, C., et al. 2014, *Exp. Astron.*, **38**, 249
- The pandas development team 2022, <https://zenodo.org/records/7344967>
- Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2015, *J. Astron. Telescopes Instrum. Syst.*, **1**, 014003
- Rodney, S. A., & Tonry, J. L. 2006, *PASP*, **118**, 866
- Rodríguez Díaz, L. F., Bigot, L., Aguirre Børsen-Koch, V., et al. 2022, *MNRAS*, **514**, 1741
- Rolland, G., Pinheiro da Silva, L., Inguibert, C., et al. 2008, *IEEE Trans. Nuclear Sci.*, **55**, 2070
- Royer, P., Clermont, L., Pertenais, M., et al. 2020, *SPIE Conf. Ser.*, **11443**, 114434Q
- Royer, P., Baeke, A., Terrasa, G., et al. 2022, *SPIE Conf. Ser.*, **12180**, 121804H
- Samadi, R., Deru, A., Reese, D., et al. 2019, *A&A*, **624**, A117
- Santin, G., Ivanchenko, V., Evans, H., Nieminen, P., & Daly, E. 2005, *IEEE Trans. Nuclear Sci.*, **52**, 2294
- Sarkar, S., Argyriou, I., Vandenbussche, B., Papageorgiou, A., & Pascale, E. 2018, *MNRAS*, **481**, 2871
- Schechter, P. L., Mateo, M., & Saha, A. 1993, *PASP*, **105**, 1342
- Short, A., Crowley, C., de Bruijne, J. H. J., & Prod'homme, T. 2013, *MNRAS*, **430**, 3078
- Silva Aguirre, V., Davsies, G. R., Basu, S., et al. 2015, *MNRAS*, **452**, 2127
- Silva Aguirre, V., Stello, D., Stokholm, A., et al. 2020, *ApJ*, **889**, L34
- Smith, J. C., Morris, R. L., Jenkins, J. M., et al. 2016, *PASP*, **128**, 124501
- Stello, D., Saunders, N., Grunblatt, S., et al. 2022, *MNRAS*, **512**, 1677
- Stetson, P. B. 1987, *PASP*, **99**, 191
- Still, M., & Barclay, T. 2012, *Astrophysics Source Code Library* [[asc1:1208.004](https://zenodo.org/records/1208004)]
- Stokholm, A., Nissen, P. E., Silva Aguirre, V., et al. 2019, *MNRAS*, **489**, 928
- Vanderburg, A., & Johnson, J. A. 2014, *PASP*, **126**, 948
- Verhove, P., Prod'homme, T., Oosterbroek, T., et al. 2016, *SPIE Conf. Ser.*, **9915**, 99150Z
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *Nat. Methods*, **17**, 261
- Walker, G., Matthews, J., Kuschnig, R., et al. 2003, *PASP*, **115**, 1023
- Wang, J., Shi, F., Zhang, J., & Liu, Y. 2008, *Pattern Recognit.*, **41**, 607
- Wehrli, C. 1985, *Extraterrestrial solar spectrum: Publication no. 615*, PMO/WRC
- Widenhorn, R., Weber-Bargioni, A., & Blouke, M. M. 2010, *Opt. Eng.*, **49**, 044401
- Zhao, X., Shen, Z., Xing, Y., & Ma, S. 2009, *Acta Aeronaut. Astronaut. Sinica*, **30**, 159
- Zima, W., Arentoft, T., De Ridder, J., et al. 2010, *ArXiv e-prints* [[arXiv:1004.0738](https://arxiv.org/abs/1004.0738)]

## Appendix A: Reference frames

This Appendix gives a detailed description of the rotation matrices in Eq. (25) which are used to determine the focal plane positions of the stars.

### A.1. The payload module reference frame

The first rotation matrix  $\mathbf{R}_{\text{EQ}}^{\text{PLM}}$  in Eq. (25) describes a passive rotation from the equatorial sky reference frame (EQ) to a reference frame associated with the payload module (PLM). Both reference frames have their origin in the geometric centre of the interface between the bottom of the optical bench and the service module. The former one is the base reference system of PlatoSim and refers to the standard right-handed celestial equatorial coordinate system  $(X_{\text{EQ}}, Y_{\text{EQ}}, Z_{\text{EQ}})$  where  $X_{\text{EQ}}$  points towards the vernal equinox, and  $Z_{\text{EQ}}$  to the equatorial north pole. We define a new right-handed Cartesian coordinate system  $(X_{\text{PLM}}, Y_{\text{PLM}}, Z_{\text{PLM}})$  which we call the payload module reference frame by performing the following rotations starting from the equatorial reference frame  $(X_{\text{EQ}}, Y_{\text{EQ}}, Z_{\text{EQ}})$ .

We first rotate over an angle  $\alpha_{\text{PLM}}$  around the  $Z_{\text{EQ}}$  axis, to obtain the  $(X_{\text{A}}, Y_{\text{A}}, Z_{\text{A}})$  reference frame where  $Z_{\text{A}} = Z_{\text{EQ}}$ . The corresponding passive rotation matrix is:

$$\mathbf{R}_{\text{EQ}}^{\text{A}} = \begin{pmatrix} \cos \alpha_{\text{PLM}} & \sin \alpha_{\text{PLM}} & 0 \\ -\sin \alpha_{\text{PLM}} & \cos \alpha_{\text{PLM}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.1})$$

Secondly, we then rotate over an angle  $\pi/2 - \delta_{\text{PLM}}$  around the  $Y_{\text{A}}$  axis, to obtain the  $(X_{\text{B}}, Y_{\text{B}}, Z_{\text{B}})$  reference frame where  $Y_{\text{B}} = Y_{\text{A}}$ . The passive rotation matrix is:

$$\mathbf{R}_{\text{A}}^{\text{B}} = \begin{pmatrix} \sin \delta_{\text{PLM}} & 0 & -\cos \delta_{\text{PLM}} \\ 0 & 1 & 0 \\ \cos \delta_{\text{PLM}} & 0 & \sin \delta_{\text{PLM}} \end{pmatrix}. \quad (\text{A.2})$$

The reason for  $\pi/2 - \delta_{\text{PLM}}$  rather than  $\delta_{\text{PLM}}$  is that  $\delta_{\text{PLM}}$  is measured from the equator rather than the north pole, while the rotation angle should increase clockwise when looking in the positive direction of the rotation axis. Lastly, we rotate over a roll angle  $\kappa_{\text{PLM}}$  around the  $Z_{\text{B}}$  axis, to obtain the  $(X_{\text{PLM}}, Y_{\text{PLM}}, Z_{\text{PLM}})$  reference frame, where  $Z_{\text{PLM}} = Z_{\text{B}}$ . The passive rotation matrix is:

$$\mathbf{R}_{\text{B}}^{\text{PLM}} = \begin{pmatrix} \cos \kappa_{\text{PLM}} & \sin \kappa_{\text{PLM}} & 0 \\ -\sin \kappa_{\text{PLM}} & \cos \kappa_{\text{PLM}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.3})$$

The first two rotations take care that the roll axis  $Z_{\text{PLM}}$  of spacecraft is pointing in the direction of the mean LOS  $(\alpha_{\text{PLM}}, \delta_{\text{PLM}})$  in the sky given by the spacecraft operator. The third rotation rolls the spacecraft so that its yaw axis  $X_{\text{PLM}}$ , which points to the highest point of the sunshield, is orientated towards the Sun.

Given a target with celestial equatorial sky coordinates  $(\alpha, \delta)$ , the Cartesian coordinates  $(v_x, v_y, v_z)_{\text{PLM}}$  of its unit direction vector in the payload module reference frame are computed using the following transformation

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}_{\text{PLM}} = \mathbf{R}_{\text{EQ}}^{\text{PLM}}(\alpha_{\text{PLM}}, \delta_{\text{PLM}}, \kappa_{\text{PLM}}) \begin{pmatrix} \cos \delta \cos \alpha \\ \cos \delta \sin \alpha \\ \sin \delta \end{pmatrix}_{\text{EQ}}, \quad (\text{A.4})$$

where the rotation matrix  $\mathbf{R}_{\text{EQ}}^{\text{PLM}}(\alpha_{\text{PLM}}, \delta_{\text{PLM}}, \kappa_{\text{PLM}})$  is given by

$$\mathbf{R}_{\text{EQ}}^{\text{PLM}}(\alpha_{\text{PLM}}, \delta_{\text{PLM}}, \kappa_{\text{PLM}}) = \mathbf{R}_{\text{B}}^{\text{PLM}} \cdot \mathbf{R}_{\text{A}}^{\text{B}} \cdot \mathbf{R}_{\text{EQ}}^{\text{A}}, \quad (\text{A.5})$$

and the rotation matrices  $\mathbf{R}_{\text{EQ}}^{\text{A}}$ ,  $\mathbf{R}_{\text{A}}^{\text{B}}$ , and  $\mathbf{R}_{\text{B}}^{\text{PLM}}$  are given by Eqs. (A.1), (A.2), and (A.3).

**Table A.1.** Overview of camera group angles.

Camera Group <sup>(a)</sup>	$\eta_{\text{CAM}}$ ( $^\circ$ ) <sup>(b)</sup>	$\rho_{\text{CAM}}$ ( $^\circ$ ) <sup>(c)</sup>
1	45	9.2
2	135	9.2
3	225	9.2
4	315	9.2
Fast	0	0.0

**Notes.** Description of each column:

(a)The camera group ID number.

(b)The azimuth rotation angle  $\eta_{\text{CAM}}$  in degrees.

(c)The tilt rotation angle  $\rho_{\text{CAM}}$  in degrees.

### A.2. The camera reference frame

The second rotation matrix  $\mathbf{R}_{\text{PLM}}^{\text{CAM}}$  in Eq. (25) describes a passive rotation from the payload module reference frame to the right-handed Cartesian camera coordinate system  $(X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})$ . It is associated with one of the cameras and has its origin exactly in the middle of the 4 CCDs in the associated focal plane. The mounting of the normal cameras on the platform is such that they are tilted and are therefore not pointing in the same direction of the pointing (roll) axis  $Z_{\text{PLM}}$  of the payload module. Instead, the optical axis  $Z_{\text{CAM}}$  of a camera is pointing towards a field-of-view that is defined by an azimuth angle  $\eta_{\text{CAM}}$  and a tilt angle  $\rho_{\text{CAM}}$  with respect to the PLM pointing. These angles are listed for the different camera groups in Table A.1. The tilt angle  $\rho_{\text{CAM}}$  of each normal camera is fixed at 9.2 degrees, and the azimuth angle is  $\eta_{\text{CAM}} = 45 + n \cdot 90$  degrees where  $n$  is the normal camera group ID number.

To define the rotation matrix  $\mathbf{R}_{\text{PLM}}^{\text{CAM}}$ , we start from the payload module reference frame  $(X_{\text{PLM}}, Y_{\text{PLM}}, Z_{\text{PLM}})$ , and perform the following rotations we rotate over a position angle  $\eta_{\text{CAM}}$  around the  $Z_{\text{PLM}}$  axis, to obtain the  $(X'_{\text{PLM}}, Y'_{\text{PLM}}, Z'_{\text{PLM}})$  reference frame where  $Z'_{\text{PLM}} = Z_{\text{PLM}}$ . The corresponding passive rotation matrix is

$$\mathbf{R}_1 = \begin{pmatrix} \cos \eta_{\text{CAM}} & \sin \eta_{\text{CAM}} & 0 \\ -\sin \eta_{\text{CAM}} & \cos \eta_{\text{CAM}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.6})$$

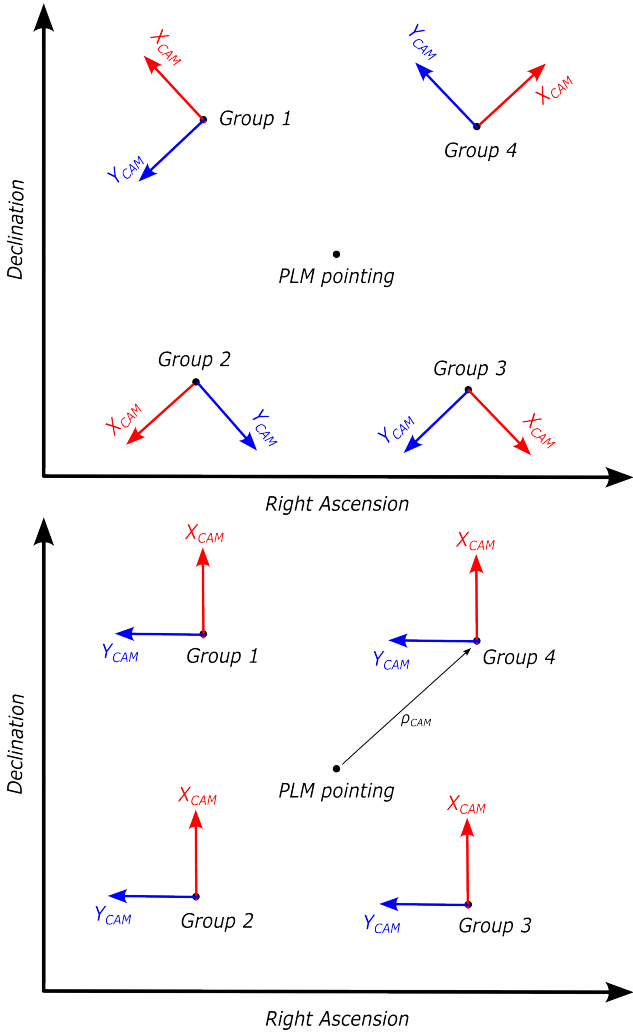
Secondly, we rotate over a tilt angle  $\rho_{\text{CAM}}$  around the  $Y'_{\text{PLM}}$  axis, to obtain the  $(X''_{\text{PLM}}, Y''_{\text{PLM}}, Z''_{\text{PLM}})$  reference frame where  $Y''_{\text{PLM}} = Y'_{\text{PLM}}$ . The corresponding passive rotation matrix is:

$$\mathbf{R}_2 = \begin{pmatrix} \cos \rho_{\text{CAM}} & 0 & -\sin \rho_{\text{CAM}} \\ 0 & 1 & 0 \\ \sin \rho_{\text{CAM}} & 0 & \cos \rho_{\text{CAM}} \end{pmatrix}; \quad (\text{A.7})$$

We finally rotate over an angle  $2\pi - \eta_{\text{CAM}}$  radians around the  $Z''_{\text{PLM}}$  axis, to obtain the  $(X_{\text{CAM}}, Y_{\text{CAM}}, Z_{\text{CAM}})$  reference frame where  $Z_{\text{CAM}} = Z''_{\text{PLM}}$ , using the passive rotation matrix

$$\mathbf{R}_3 = \begin{pmatrix} \cos \eta_{\text{CAM}} & -\sin \eta_{\text{CAM}} & 0 \\ \sin \eta_{\text{CAM}} & \cos \eta_{\text{CAM}} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \mathbf{R}'_1. \quad (\text{A.8})$$

The first two rotations serve to point the camera on the platform to the line-of-sight defined by the two angles  $\eta_{\text{CAM}}$  and  $\rho_{\text{CAM}}$ . The last rotation takes care that the  $X_{\text{CAM}}$  and  $Y_{\text{CAM}}$  axes of the different camera groups are aligned. For example, without the extra rotation, the location of the spacecraft pointing axis  $(x_{\text{PLM}}, y_{\text{PLM}}, z_{\text{PLM}}) = (0, 0, 1)$  in the camera reference frame would be  $(x_{\text{CAM}}, y_{\text{CAM}}, z_{\text{CAM}}) = (-\sin \varrho, 0, \cos \varrho)$  (cf. Eq. (A.9)). That is,



**Fig. A.1.** Reference frames in equatorial coordinates. *Top*: the orientation of the camera reference frame for each of the camera groups when there would be no extra rotation over an angle  $-\eta_{\text{CAM}}$ . *Bottom*: the orientation of the same reference frames with the extra rotation (A.8).  $\rho_{\text{CAM}}$  is the camera tilt angle. The right-hand side orientation allows for nearly aligned camera axes  $X_{\text{CAM}}$  and  $Y_{\text{CAM}}$  of the different camera groups on the sky.

in the camera  $xy$ -plane the payload module pointing axis would cross the negative  $x_{\text{CAM}}$ -axis, independent of the camera group. This is illustrated in Fig. A.1. The extra rotation allows to align the focal planes of the different camera groups in the sky, and has the additional benefit that the payload module pointing coordinates always fall on (rather than in between) a CCD for each of the 4 camera groups.

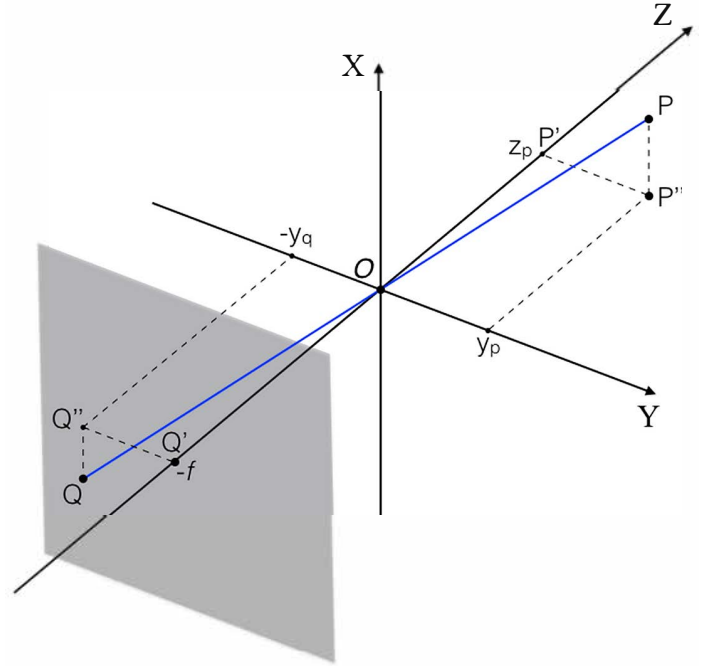
A vector with coordinates  $(x_{\text{PLM}}, y_{\text{PLM}}, z_{\text{PLM}})$  in the payload module reference frame, has coordinates  $(x_{\text{CAM}}, y_{\text{CAM}}, z_{\text{CAM}})$  in the camera reference frame that can be computed using the following transformation:

$$\begin{pmatrix} x_{\text{CAM}} \\ y_{\text{CAM}} \\ z_{\text{CAM}} \end{pmatrix} = \mathbf{R}_{\text{PLM}}^{\text{CAM}}(\eta_{\text{CAM}}, \rho_{\text{CAM}}) \begin{pmatrix} x_{\text{PLM}} \\ y_{\text{PLM}} \\ z_{\text{PLM}} \end{pmatrix}, \quad (\text{A.9})$$

where

$$\mathbf{R}_{\text{PLM}}^{\text{CAM}}(\eta_{\text{CAM}}, \rho_{\text{CAM}}) = \mathbf{R}_1^t \cdot \mathbf{R}_2 \cdot \mathbf{R}_1, \quad (\text{A.10})$$

where we used the rotation matrices (A.6), (A.7) and (A.8) enumerated above.



**Fig. A.2.** Standard pinhole camera model. The  $Z$  axis corresponds to the optical axis of the camera. The origin of the reference frame is the pinhole  $O$ , the focal plane is located behind the pinhole at a distance  $f$  where  $f$  is the focal length of the camera. Light rays of a star at point  $P$ , go through the pin hole  $O$  and are projected to a point  $Q$  on the focal plane.

### A.3. The focal plane reference frame

The third rotation matrix  $\mathbf{R}_{\text{CAM}}^{\text{FP}}$  in Eq. (25) describes the projection of the sky on the focal plane of a camera. To do so, we use the standard pinhole camera model as is shown in Fig. A.2. The  $Z$  axis is the optical axis of the camera, the  $(X, Y)$  plane is parallel to the focal plane, but has its origin in the pinhole. The focal plane (with the CCDs) is at a distance  $f$  behind the pinhole, where  $f$  is the focal length of the camera. Light rays of a star at point  $P$  go through the pin hole  $O$  and are projected to a point  $Q$  on the focal plane. In the  $(Y, Z)$  plane, the triangles  $OP'P''$  and  $OQ'Q''$  are similar triangles, so that we have  $|P'P''|/|OP'| = |Q'Q''|/|OQ'|$  or

$$\frac{-y_q}{f} = \frac{y_p}{z_p}, \quad (\text{A.11})$$

where the minus sign indicates that the pinhole reverses the image on the focal plane. A similar argument holds in the  $(X, Z)$  plane, so that we can write

$$\begin{pmatrix} x_q \\ y_q \\ -f \end{pmatrix} = -\frac{f}{z_p} \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}. \quad (\text{A.12})$$

With normal coordinates, the non-linear division by  $z_p$  prevents expressing the pinhole projection as a convenient matrix multiplication. We therefore go to augmented space and use homogeneous coordinates  $(x_q, y_q, 1)$  and  $(x_p, y_p, z_p, 1)$

$$\begin{pmatrix} x_q \\ y_q \\ z_q \\ 1 \end{pmatrix} \sim \mathbf{C} \cdot \begin{pmatrix} x_p \\ y_p \\ z_q \\ 1 \end{pmatrix} \quad (\text{A.13})$$

where the equality = has been replaced by the equivalence relation  $\sim$ . After any transformation a homogenisation of the coordinates is executed, which means dividing all coordinates by the fourth one. The camera matrix  $\mathbf{C}$  used by PlatoSim is

$$\mathbf{C} = \begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & -f & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (\text{A.14})$$

which mirrors the image after the pinhole projection.

We can now define the right-handed Cartesian coordinate system  $(X_{\text{FP}}, Y_{\text{FP}}, Z_{\text{FP}})$  associated with the focal plane of one camera with its origin exactly in the middle of the 4 CCDs (see Fig. 3 for an illustration). The  $Z_{\text{FP}}$  axis coincides with the  $Z_{\text{CAM}}$  optical axis of the camera, which points towards the FOV. The focal plane  $(X_{\text{FP}}, Y_{\text{FP}})$  is obtained by rotating the  $(X_{\text{CAM}}, Y_{\text{CAM}})$  plane over an angle  $\gamma_{\text{FP}}$  called the focal plane orientation around the  $Z_{\text{FP}} = Z_{\text{CAM}}$  axis. The corresponding transformation (using homogeneous coordinates) is

$$\begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \\ -f \\ 1 \end{pmatrix} = \mathbf{R}_{\text{CAM}}^{\text{FP}}(\gamma_{\text{FP}}) \begin{pmatrix} x_{\text{CAM}} \\ y_{\text{CAM}} \\ z_{\text{CAM}} \\ 1 \end{pmatrix}, \quad (\text{A.15})$$

where the rotation matrix  $\mathbf{R}_{\text{CAM}}^{\text{FP}}(\gamma_{\text{FP}})$  is given by:

$$\mathbf{R}_{\text{CAM}}^{\text{FP}}(\gamma_{\text{FP}}) = \begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & -f & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos \gamma_{\text{FP}} & \sin \gamma_{\text{FP}} & 0 & 0 \\ -\sin \gamma_{\text{FP}} & \cos \gamma_{\text{FP}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{A.16})$$

Here  $(x_{\text{CAM}}, y_{\text{CAM}}, z_{\text{CAM}}, 1)$  are the homogenised coordinates of a vector in the camera reference frame, and  $(x_{\text{FP}}, y_{\text{FP}}, -f, 1)$  are the coordinates of the homogeneous projected vector in the focal plane reference frame. We note that  $\mathbf{R}_{\text{CAM}}^{\text{FP}}$  is not invertible: projecting loses the  $z_{\text{CAM}}$  information. However, since PlatoSim only uses (sky) directions, the reverse transformation can be obtained by enforcing the length of the vector in the camera reference frame to be unity.

#### A.4. The CCD reference frame

The fourth transformation matrix  $\mathbf{R}_{\text{FP}}^{\text{CCD}}$  in Eq. (25) describes a passive rotation from the focal plane reference frame to the right-handed Cartesian CCD coordinate system  $(X_{\text{CCD}}, Y_{\text{CCD}})$ , which is associated with one of the CCDs in a camera. Its origin is in one of the corners of the CCD. The charge transfer direction during readout of the CCD happens in the negative  $Y_{\text{CCD}}$  direction towards  $y_{\text{CCD}} = 0$ , and the readout of the serial readout register happens in the negative  $X_{\text{CCD}}$  direction (towards the origin) for the left side of the CCD, and in the opposite direction for the right side of the CCD. The orientation of the CCD with respect to the  $(X_{\text{FP}}, Y_{\text{FP}})$  coordinate system is defined by the angle  $\gamma_{\text{CCD}}$ , as shown in Fig. 3. The transformation (in homogeneous coordinates, including a rotation as well as a translation) between the focal plane and the CCD reference frames is implemented by

$$\begin{pmatrix} x_{\text{CCD}} \\ y_{\text{CCD}} \\ 1 \end{pmatrix} = \mathbf{R}_{\text{FP}}^{\text{CCD}}(\gamma_{\text{CCD}}) \cdot \begin{pmatrix} x_{\text{FP}} \\ y_{\text{FP}} \\ 1 \end{pmatrix}, \quad (\text{A.17})$$

**Table A.2.** Overview of the CCD reference frame parameters.

CCD <sup>(a)</sup> ID	$N_{\text{rows}}$ <sup>(b)</sup> (pix)	$N_{\text{cols}}$ <sup>(b)</sup> (pix)	$r_{\text{exp}}$ <sup>(c)</sup> (pix)	$\Delta x_{\text{CCD}}$ <sup>(d)</sup> (mm)	$\Delta y_{\text{CCD}}$ <sup>(d)</sup> (mm)	$\gamma_{\text{CCD}}$ <sup>(e)</sup> (degree)
1	4510	4510	0	-1.3	82.48	180
2	4510	4510	0	-1.3	82.48	270
3	4510	4510	0	-1.3	82.48	0
4	4510	4510	0	-1.3	82.48	90
1 (F)	4510	4510	2255	-1.3	82.48	180
2 (F)	4510	4510	2255	-1.3	82.48	270
3 (F)	4510	4510	2255	-1.3	82.48	0
4 (F)	4510	4510	2255	-1.3	82.48	90

**Notes.** Description of each column:

(a) The CCD codes 1, 2, 3, and 4 for a N-CAM, 1 (F), 2 (F), 3 (F), and 4 (F) for a F-CAM.

(b) The number of rows and columns for each CCD.

(c) First row that is exposed; for the fast camera only rows 2255 until 4510 are exposed.

(d) The CCD origin coordinates in the  $(X_{\text{FP}}, Y_{\text{FP}})$  frame.

(e) The CCD rotation angle.

where

$$\mathbf{R}_{\text{FP}}^{\text{CCD}}(\gamma_{\text{CCD}}) = \begin{pmatrix} \cos \gamma_{\text{CCD}} & \sin \gamma_{\text{CCD}} & \Delta x_{\text{CCD}} \\ -\sin \gamma_{\text{CCD}} & \cos \gamma_{\text{CCD}} & \Delta y_{\text{CCD}} \\ 0 & 0 & 1 \end{pmatrix}, \quad (\text{A.18})$$

and where  $(\Delta x_{\text{CCD}}, \Delta y_{\text{CCD}})$  are the coordinates of the CCD zeropoint in the  $(X_{\text{FP}}, Y_{\text{FP}})$  frame. Equation (A.17) allows to use the same  $\Delta x_{\text{CCD}}$  and  $\Delta y_{\text{CCD}}$  for each CCD, independent of the CCD rotation angle  $\gamma_{\text{CCD}}$ . An overview of the zero point coordinates and the orientation angle  $\gamma_{\text{CCD}}$  is given in Table A.2. PlatoSim chooses to define the Focal Plane reference frame such that its axes run parallel with the CCD axes of CCD #3.

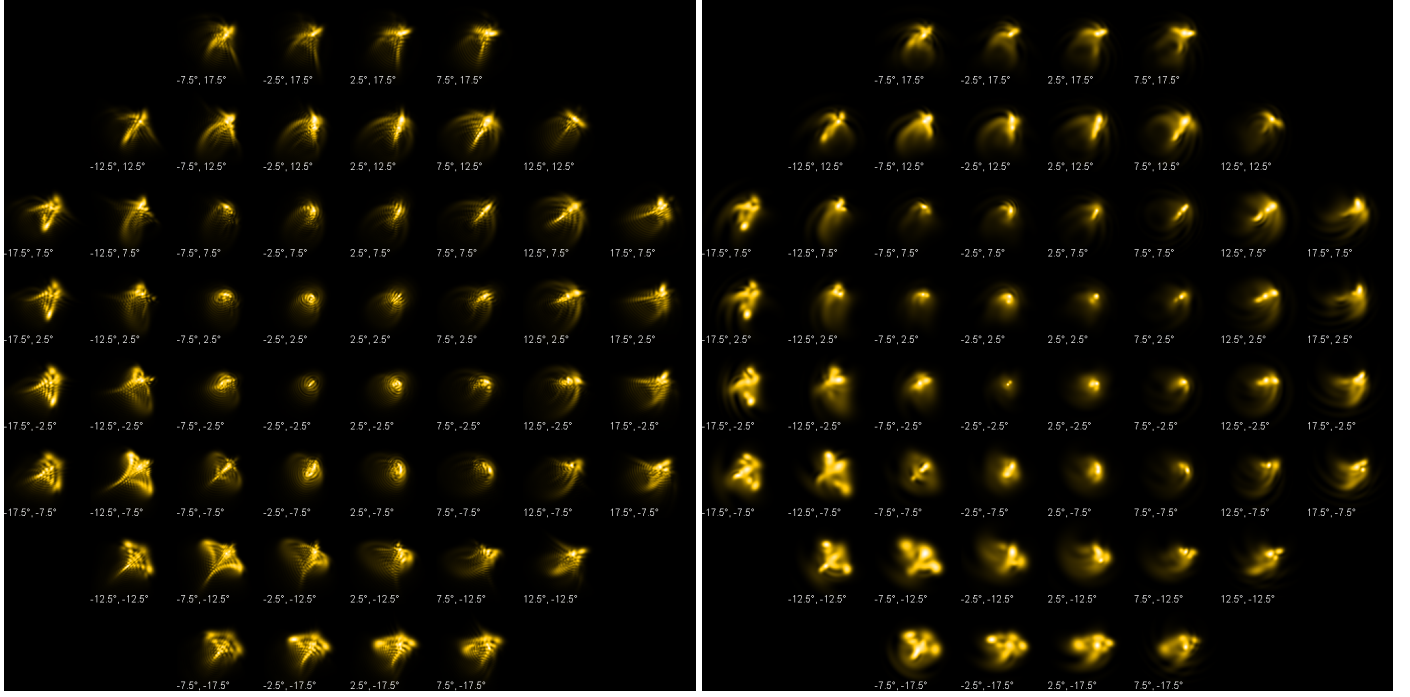
## Appendix B: The PLATO PSF

As for most cameras, the PSF depends on the angular radial distance between the barycentre of the PSF in the focal plane  $(x, y)$  to the optical axis. The latter is assumed to go through the origin  $(0, 0)$  of the focal plane. This angular distance  $\vartheta$  can be computed in the focal plane reference frame with

$$\cos \vartheta = \left[ 1 + \left( \frac{x_{\text{FP}}}{f} \right)^2 + \left( \frac{y_{\text{FP}}}{f} \right)^2 \right]^{-1/2}, \quad (\text{B.1})$$

where  $(x_{\text{FP}}, y_{\text{FP}})$  are the pinhole-projected focal plane coordinates of the centre of the PSF (cf. Sect. A.3). Moreover, the exact morphology of the PSF is not only dependent on  $\vartheta$  but also on the orientation angle  $\theta_{\text{OA}}$  of the PSF. A PSF therefore has to be properly rotated at its location in the focal plane.

PlatoSim has two options for incorporating PSFs: synthetic and analytical ones. As part of the mission preparation, synthetic PSFs of the PLATO cameras have been extensively generated with the ray-tracing software package Zemax for many different positions in the focal plane, for both in-focus and out-of-focus PSFs, and for different stellar effective temperatures. For a given subfield with point sources in the focal plane, the optimal PSF corresponding to the subfield centre is selected from the library, and applied to the entire subfield using a convolution with a Fast Fourier Transform. As a result, with this option the PSF is the



**Fig. B.1.** Normalised non-diffused PLATO PSFs. *Left:* PSFs generated with Zemax OpticStudio. *Right:* PSFs generated with the analytic model. Both plots display the PSF across the full FPA and the focal plane coordinate is shown below the corresponding PSF.

same across the complete subfield. Being a compromise of computational cost, the assumption of a constant PSF becomes less valid with increasing subfield size.

In the case of analytic PSF models a parametric model is used, which is defined as

$$G_N(x, y) = \frac{1}{Z} \left( \sum_{n=1}^{N_G} f_{G,n}(x, y) + \sum_{l=1}^{N_S} f_{S,l}(x, y) + \sum_{k=1}^{N_C} f_{C,k}(x, y) \right), \quad (\text{B.2})$$

where the three base functions are given by

$$\begin{aligned} f_G(x, y) &= A e^{-\beta_0}, \\ f_S(x, y) &= A e^{-\beta_0} \sin^2(\delta \beta_s), \\ f_C(x, y) &= A e^{-\beta_0} \cos^2(\delta \beta_s), \end{aligned} \quad (\text{B.3})$$

where  $\sigma$  is the generalised standard deviation of the PSF,  $\delta$  describes the morphology of the PSF side lobes, and  $(\beta_0, \beta_s)$  are abbreviations for the following expressions:

$$\begin{aligned} \beta_0 &\equiv \frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}, \\ \beta_s &\equiv \frac{(x + x_s - x_0)^2 + (y + y_s - y_0)^2}{4\sigma^2}. \end{aligned}$$

The normalisation constant  $Z$  in Eq. (B.2) is given by

$$Z = \sum_{n=1}^{N_G} Z_{G,n} + \sum_{l=1}^{N_S} Z_{S,l} + \sum_{k=1}^{N_C} Z_{C,k}, \quad (\text{B.4})$$

where

$$\begin{aligned} Z_G &= 2 A \pi \sigma^2, \\ Z_S &= A \pi \sigma^2 (1 - \kappa_0), \\ Z_C &= A \pi \sigma^2 (1 + \kappa_0), \end{aligned} \quad (\text{B.5})$$

with  $\kappa_0$  and  $\kappa_s$  defined as

$$\begin{aligned} \kappa_0 &\equiv \frac{e^{-\delta \kappa_s} (\cos \kappa_s - \delta \sin \kappa_s)}{1 + \delta^2}, \\ \kappa_s &\equiv \frac{\delta (x_s^2 + y_s^2)}{2\sigma^2(1 + \delta^2)}. \end{aligned}$$

We note that  $\lim_{\delta \rightarrow 0} f_C = f_G$  and that if two base function  $f_S$  and  $f_C$  have the same parameters  $(x_0, y_0, A, \sigma, \delta, x_s, y_s)$  their sum  $f_S + f_C = f_G$ .

The advantage of this parametric model is that it allows for a wide variety of PSF morphologies, and that it can be efficiently integrated over a pixel. For example, in the case of the base function  $f_G(x, y)$  we have

$$\int_{x_i}^{x_{i+1}} \int_{y_i}^{y_{i+1}} \frac{1}{Z_G} f_G(x, y) dx dy = \frac{1}{4} \mathcal{E}_x \mathcal{E}_y, \quad (\text{B.6})$$

where  $\mathcal{E}_x$  and  $\mathcal{E}_y$  are given by

$$\begin{aligned} \mathcal{E}_x &= \operatorname{erf} \left( \frac{x_{i+1} - x_0}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left( \frac{x_i - x_0}{\sqrt{2}\sigma} \right), \\ \mathcal{E}_y &= \operatorname{erf} \left( \frac{y_{i+1} - y_0}{\sqrt{2}\sigma} \right) - \operatorname{erf} \left( \frac{y_i - y_0}{\sqrt{2}\sigma} \right). \end{aligned} \quad (\text{B.7})$$

Here  $\operatorname{erf}(x)$  is the standard error function given by

$$\operatorname{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (\text{B.8})$$

which can be very efficiently computed using the Faddeeva<sup>16</sup> package.

<sup>16</sup> [http://ab-initio.mit.edu/wiki/index.php/Faddeeva\\_Package](http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package)

In practice, the model parameters ( $A, \sigma, \delta, x_s, y_s$ ) need to be fitted, which was done using the Zemax PSFs as a reference model. The metric used was the sum of the absolute values of all subpixel value differences between the Zemax PSF and the analytic PSF. This led to a typical residual deviation between the two of  $\sim 12\%$  for the PSFs without charge diffusion, and  $\sim 6\%$  for PSFs with charge diffusion.

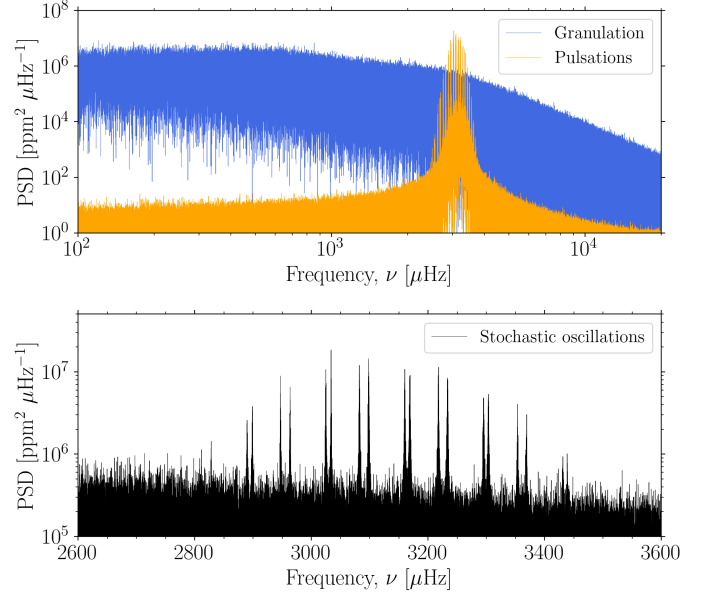
The analytic PSFs are not convolved, but simply summed over the subfield. This has the advantage that the PSF variation over the focal plane can be taken into account, which is particular relevant for a large subfield.

For a comparison, Fig. B.1 shows the PSF across the focal plane modelled with Zemax (left) and the analytic (right) one. The relatively large PLATO plate scale of  $15 \text{ arcsec pixel}^{-1}$  means that on average across the FOV 99% of the PSF flux is enclosed in a subfield of  $5 \times 5$  pixels. The sparse sampling of the PSF makes the flux distribution of a star very sensitive to its barycentre location within a pixel (see Sec. 9). Furthermore, asymmetric PSFs that contains large spikes (more typical for larger angular distance from the optical axis) further enhance effects of pixel-to-pixel and intra-pixel sensitivities as one or more pointing errors moves the PSF relative to the pixel array.

### Appendix C: Stochastic oscillations

Since both the P1, P2, and P5 samples of the asPIC comprise stars of spectral type F5-K7 (Montalto et al. 2021), we expect all stars to show photometric variability due to granulation and stochastic oscillations. To efficiently model and include these signals in simulations with PLATOsim, BiSON observations<sup>17</sup> of 96 distinct pulsation mode frequencies of the Sun were used as a benchmark. We model the frequency of maximum power ( $\nu_{\max}$ ) and the primary frequency spacing ( $\Delta\nu$ ) from the asteroseismic scaling relation of Kjeldsen & Bedding (1995). The granulation signal component is modelled using 2 super-Lorentzian functions as per Kallinger et al. (2014) and the pulsation amplitudes are modelled using the methodology of Corsaro et al. (2013) (which use *Kepler* red giant stars in the open clusters NGC 6791 and NGC 6819, together with a sample of main sequence and sub-giant *Kepler* field stars). The granulation and pulsation amplitude spectrum is rescaled to the PLATO pass-band outlined in Sarkar et al. (2018) (initially developed for the ARIEL mission), with the general methodology structured around determining the bolometric coefficient from an interpolation of synthetic spectra – here we use the PHOENIX library<sup>18</sup> (Husser et al. 2013). Compared to Sarkar et al. (2018)’s use of the PHOENIX NextGen library of Atmospheric models, we here use the PHOENIX high resolution spectral models for a more complete space of parameters  $\{M_s, R_s, T_{\text{eff}}, \log g, Z, \alpha\}$ .

We simulate time series of stochastically damped modes using the formalism described in De Ridder et al. (2006). Being equivalent to the traditional formalism of modelling solar-like oscillations in the Fourier domain (Anderson et al. 1990) the code produces time series directly from the time domain. We emphasise that the applied formalism (and the one formulated by Anderson et al. 1990) are simplistic models which do not take into account a full physical description of turbulent convection nor rotation. Nevertheless, comparing our results using a solar-like analogue, we note that our methodology results in a granulation amplitude of  $\sim 48.9$  ppm which is higher than the value reported by Kallinger et al. (2016) of 41(2) ppm. However,



**Fig. C.1.** Power spectral density (PSD) plots of a 2 yr simulated light curve due to convection-driven variability. *Top:* Components of the simulated granulation (blue) and stochastic oscillations (orange). *Bottom:* Zoom-in on the granulation and pulsation model clearly showing the recognisable envelope of frequency peaks (used to estimate  $\nu_{\max}$ ) and the small ( $\delta\nu$ ) and large ( $\Delta\nu$ ) frequency separations.

it resembles more closely the value of 46 ppm from a recent study by Rodríguez Díaz et al. (2022) which uses 3D stellar atmosphere models. Regarding the pulsation amplitude we find that our value of 10.6 ppm is in good agreement with the result from Kallinger et al. (2016) of 11(6) ppm.

<sup>17</sup> Main\_Fits\_BiSON\_8640d\_lbest\_UseInSolarCycle.dat

<sup>18</sup> <https://phoenix.astro.physik.uni-goettingen.de/>