Entwicklung eines modularen Belichtungszeitrechners für ballonbasierte Teleskope

Development of a modular exposure time calculator for balloon-based telescopes

Masterarbeit von
B. Sc. Lukas Klass
IRS-20-S-081

Betreuer:
Prof. Dr. rer. nat. Alfred Krabbe
Dipl.-Ing. Philipp Maier

Institut für Raumfahrtsysteme, Universität Stuttgart
November 2020

UNIVERSITÄT STUTTGART
INSTITUTE OF SPACE SYSTEMS

Pfaffenwaldring 29 · 70569 Stuttgart · Phone +49 (0) 711 685-62375 · Fax: +49 (0) 711 685-63596 · www.irs.uni-stuttgart.de

Master Thesis Work

of Mr. Klass, Lukas

Entwicklung eines modularen Belichtungszeitrechners für ballonbasierte Teleskope

Development of a modular exposure time calculator for balloon-based telescopes

Motivation:

Observations in the far-infrared spectral range between approx. 30 and 300 µm are essential for answering open questions in modern astronomy, including such about the origin and development of complex matter as we know it. However, due to the almost complete absorption of far infrared light in the atmosphere, observations in this region cannot be carried out from the ground.

A possible alternative to comparatively costly satellite and airborne observatories is the use of telescopes carried by scientific stratospheric balloons. These can, compared to airborne and satellite observatories, support large telescopes relatively easily and thereby enable high spatial resolution. Furthermore, stratospheric balloons can reach flight altitudes of 30 to 40 km, overcoming 99 % of the otherwise light-absorbing atmospheric mass, particularly including the atmospheric water vapor. In order to enable a detailed estimation of the expectable performance of balloon telescopes under different flight scenarios, this master thesis shall develop an exposure time calculator for observations in the far infrared. In order to allow further development, expansion, and the addition of new instruments in the future, the estimator shall be written as modular software in python. The estimator shall then be used exemplarily to determine the expected performance of large aperture balloon telescopes to a current science case in the far infrared, the detection of light hydrides.

Task:

After detailed familiarization with the topic, the following tasks shall be performed:

- Definition of a concept for a modular exposure time calculator
- Implementation of a core exposure time calculator for balloon-based telescopes in Python
- Definition and execution of verification tests for the core exposure time calculator
- Implementation of a far-infrared heterodyne spectrometer as part of the exposure time calculator
- Verification of the heterodyne spectrometer module
- Analysis of the expected performance of a balloon-based heterodyne spectrometer for a selected science application (high spectral resolution observation of light hydrides)

Supervisor: Dipl.-Ing. Philipp Maier

Starting date: 01.05.2020

Submission until: 20.12.2020

Acknowledgement of receipt:
I hereby confirm that I read and understood the task of the bachelor thesis, the juridical regulations as well as the study- and exam regulations.

Prof. Dr. rer. nat. A. Krabbe
(Responsible Professor)

Signature of the student

Declaration

I, ***Klass, Lukas*** hereby certify that I have written this ***Master thesis*** independently with the support of the supervisor, and I did not use any resources apart from those specified. The thesis, or substantial components of it, has not been submitted as part of graded course work at this or any other educational institution.

I also declare that during the preparation of this thesis I have followed the appropriate regulations regarding copyright for the use of external content, according to the rules of good scientific and academic practice[1]. I have included unambiguous references for any external content (such as images, drawings, text passages etc.), and in cases for which approval is required for the use of this material, I have obtained the approval of the owner for the use of this content in my thesis. I am aware that I am responsible in the case of conscious negligence of these responsibilities.

Stuttgart, 18.11.2020, L.Kl

Place, Date, Sign

I hereby agree that my ***Master thesis*** with the following title:

Development of a modular exposure time calculator for balloon-based telescopes

is archived and publicly available in the library of the Institute of Space Systems of the University of Stuttgart ***without blocking period*** and that the thesis is available on the website of the institute as well as in the online catalogue of the library of the University of Stuttgart. The latter means that bibliographic data of the thesis (title, author, year of publication, etc.) is permanently and worldwide available.

After finishing the work I will, for this purpose, deliver a further copy of the thesis along with the examination copy, as well as a digital version.

I transfer the proprietary of these additional copies to the University of Stuttgart. I concede that the thesis and the results generated within the scope of this work can be used free of cost and of temporal and geographical restrictions for the purpose of research and teaching to the institute of Space Systems. If there exist utilisation right agreements related to the thesis from the institute or third parties, then these agreements also apply for the results developed in the scope of this thesis.

Stuttgart, 18.11.2020, L.Kl

Place, Date, Sign

[1] Stated in the DFG recommendations for „Assurance of Good Scientific Practice" or in the statute of the University of Stuttgart for „Ensuring the Integrity of Scientific Practice and the Handling of Misconduct in Science"

Contents

Abstract

The development of ESBO, the planned European Stratospheric Balloon Observatory, requires a tool for the calculation of the observatory's potential performance. This tool is intended to support the selection of the best possible configuration of telescope and instruments as well as the optimal components. A key indicator for an observatory's performance is the exposure time required for a given signal-to-noise ratio. Later, during the operation of the observatory, the required exposure time is of importance for planning the scientific observations.

For this reason, a modular exposure time calculator, which allows to calculate the performance of the observatory by using different components will be developed in this master thesis. Furthermore, this software can be later used by scientists to calculate the necessary observation time for their science cases. Therefore the exposure time calculator offers an extensive library of components that can be configured and combined via a configuration file to model the optical path. Inter alia, this library contains different sources like black body radiators, optical components like the atmosphere, mirrors or lenses, and two types of detectors. A special focus of this thesis is the modeling and implementation of a heterodyne detector that can be used for high spectral resolution line observations.
At first, the physical foundation for these calculations is developed by deriving the appropriate equations.
An object-oriented software architecture ensures that future components can be easily added. All available components as well as the associated parameters, the code and the corresponding interfaces are explained in detail in a documentation.
To ensure the correct operation of the exposure time calculator, individual module tests are implemented during the development phase. After completion, the entire software is then fully verified using other exposure time calculators and hand calculations.
Finally, the developed exposure time calculator is applied to two science cases to validate the maturity of the software. First, an observation with SOFIA/FORECAST is simulated and compared to the SOFIA exposure time calculator. The same simulation is then carried out with improved mirror parameters to determine their influence on the required exposure time. In a second science case, the observation of an absorption line of HCl^+ towards W31C with ESBO and a heterodyne instrument, as it was already performed in 2012 with Herschel/HIFI, is simulated and compared. It is shown that ESBO can reach almost the performance of the Herschel space observatory.

Zusammenfassung

Für die Entwicklung von ESBO, dem geplanten *European Stratospheric Balloon Observatory*, wird ein Werkzeug für die Berechnung der potentiellen Leistungsfähigkeit des Observatoriums benötigt. Dieses soll die Auswahl der bestmöglichen Konfiguration von Teleskop und Instrumente sowie der optimalen Komponenten unterstützen. Eine gute Kenngröße für die Leistungsfähigkeit eines Observatoriums stellt die für ein bestimmtes Signal-Rausch-Verhältnis notwendige Belichtungszeit dar. Diese ist auch später während des Betriebs des Observatoriums für die Planung der wissenschaftlichen Beobachtungen von Bedeutung.

Aus diesem Grund wird im Rahmen dieser Masterarbeit ein modularer Belichtungszeitrechner entwickelt, der es ermöglicht, die Leistungsfähigkeit des Observatoriums unter der Verwendung verschiedener Komponenten zu berechnen. Darüber hinaus kann dieser später von Wissenschaftlern verwendet werden, um die notwendige Beobachtungszeit für ihre wissenschaftlichen Fragestellungen zu berechnen. Hierfür bietet der Belichtungszeitrechner eine umfangreiche Bibliothek an Komponenten. Über eine Konfigurationsdatei können diese konfiguriert und kombiniert werden, um damit den optischen Pfad nachzubilden. Unter anderem beinhaltet diese Bibliothek verschiedene Quellen wie Schwarzkörperstrahler, optische Komponenten wie die Atmosphäre, Spiegel oder Linsen, sowie zwei Arten von Detektoren. Ein besonderer Schwerpunkt der Arbeit stellt die Modellierung und Implementierung eines Heterodyn-Instruments dar, das für spektrale hochauflösende Linien-Beobachtungen verwendet werden kann.
Zunächst wird deshalb die hierfür notwendigen physikalische Grundlage für die Berechnungen geschaffen indem die entsprechenden Zusammenhänge hergeleitet werden.
Eine objekt-orientierte Architektur der Software stellt sicher, dass auch später weitere Komponenten hinzugefügt werden können. In einer Dokumentation sind sowohl alle verfügbaren Komponenten als auch die zugehörigen Parameter sowie der Code und die entsprechenden Schnittstellen ausführlich erläutert.
Um die korrekte Funktion des Belichtungszeitrechners zu gewährleisten, werden bereits während der Entwicklung einzelne Modultests implementiert. Nach der Fertigstellung wird dann die gesamte Software mithilfe anderer Belichtungszeitrechner und Handrechnungen vollständig verifiziert.
Abschließend wird der fertige Belichtungszeitrechner auf zwei wissenschaftliche Fragestellungen angewendet, um damit die Reife der Software zu validieren. Zunächst wird eine Beobachtung mit SOFIA/FORECAST simuliert und mit dem SOFIA Belichtungszeitrechner verglichen. Dieselbe Simulation wird anschließend mit verbesserten Spiegelparametern durchgeführt um deren Einfluss auf die Benötigte Belichtungszeit zu bestimmen.
In einer zweiten Fragestellung wird die Beobachtung einer Absorptionslinie von HCl^+ mit ESBO und einem Heterodyn-Instrument, wie sie bereits 2012 mit Herschel/HIFI durchgeführt wurde, simuliert und verglichen. Es zeigt sich, dass ESBO beinahe die Leistungsfähigkeit des Weltraumobservatoriums Herschel erreichen kann.

Acronyms

AETC	Advanced Exposure Time Calculator
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CCD	charge coupled device
CI/CD	continuous integration / continuous development
CSV	comma separated values
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DSB	dual sideband
DSI	Deutsches SOFIA Institut
EM	electromagnetic
ESBO	European Stratospheric Balloon Observatory
ESBO-*DS*	European Stratospheric Balloon Observatory – Design Study
ETC	exposure time calculator
FIR	far-infrared
FITS	Flexible Image Transport System
FORECAST	Faint Object Infrared Camera for the SOFIA Telescope
FOV	field of view
FWHM	full width half maximum
GREAT	German Receiver for Astronomy at Terahertz Frequencies
GUI	graphical user interface
H.O.T.	higher order terms
HIFI	Heterodyne Instrument for the Far Infrared
HSA	Herschel Science Archive
HTML	Hypertext Markup Language
IF	intermediate frequency
JSON	JavaScript Object Notation
LO	local oscillator
LSB	lower sideband
NIR	near-infrared
OOP	object-oriented programming
PACS	Photodetector Array Camera and Spectrometer
PI	principle investigator
PRISMAS	Probing InterStellar Molecules with Absorption line Studies
PSD	power spectral density
PSF	point spread function

RMS	root mean square
SITE	SOFIA Instrument Time Estimator
SNR	signal-to-noise ratio
SOFIA	Stratospheric Observatory for Infrared Astronomy
SSB	single sideband
STUDIO	Stratospheric Ultraviolet Demonstrator of an Imaging Observatory
UML	Unified Modeling Language
USB	upper sideband
UV	ultraviolet
XML	Extensible Markup Language

Symbols

Symbol	Description	Unit
A	Area	m^2
A_{ap}	Aperture area	m^2
A_{eff}	Effective antenna area	m^2
A_{obs}	Obstructor area	m^2
c	Speed of light	$\mathrm{m\,s}^{-1}$
D_{ap}	Aperture diameter	m
D_{phot}	Photometric aperture diameter	m
e	Euler's number	-
E	Radiant flux density	$\mathrm{W\,m}^{-2}$
E_ν	Spectral flux density	$\mathrm{W\,m}^{-2}\,\mathrm{Hz}^{-1}$
EE	Percentage of encircled energy	%
f	Telescope's focal length	m
F	Antenna pattern	-
h	Planck constant	$\mathrm{J\,s}$
I	PSF intensity	-
I_D	Dark current	$\mathrm{e^-\,s}^{-1}$
k_B	Boltzmann constant	$\mathrm{J\,K}^{-1}$
l	Pixel size	m
L_Ω	Radiance	$\mathrm{W\,m}^{-2}\,\mathrm{sr}^{-1}$
$L_{\Omega,\nu}$	Spectral radiance	$\mathrm{W\,m}^{-2}\,\mathrm{sr}^{-1}\,\mathrm{Hz}^{-1}$
m	Apparent magnitude	mag
n_{osf}	Oversampling factor	-
N	Telescope's focal number	-
N_{e^-}	Number of electrons	$\mathrm{e^-}$
N_{pix}	Number of pixels	pixel
N_{phot}	Number of photons	photon
$N_{phot,\nu}$	Spectral number of photons	$\mathrm{photon\,Hz}^{-1}$
o	Quadratic aperture obstruction ratio	-
P	Power	W
P_ν	Power spectral density	$\mathrm{W\,Hz}^{-1}$
Q	Radiant energy	J
QE	Quantum efficiency	$\mathrm{e^-\,photon}^{-1}$
QE_ν	Spectral quantum efficiency	$\mathrm{e^-\,photon}^{-1}\,\mathrm{Hz}^{-1}$
r	Line of sight	m
SNR	Signal-to-noise ratio	-
S	Surface brightness	$\mathrm{mag/sr}^2$
t	Time	s
t_{exp}	Exposure time	s

Symbol	Description	Unit
T	Temperature	K
T_A	Antenna temperature	K
T_{bg}	Background temperature	K
T_B	Brightness temperature	K
T_{mb}	Main beam temperature	K
T_{rec}	Receiver temperature	K
ΔT_{rms}	RMS antenna temperature	K
T_{sys}	System temperature	K
α	Absorption coefficient	-
α_ν	Spectral absorption coefficient	Hz^{-1}
η_{ap}	Aperture efficiency	-
η_{fss}	Forwards scattering efficiency	-
η_{mb}	Main beam efficiency	-
θ	Angle of incidence	rad
$\bar{\theta}$	Reduced observation angle	rad
κ	Back-end degradation factor	-
λ	Wavelength	m
ν	Frequency	Hz
$\Delta\nu$	Bandwidth	Hz
ρ	Reflection coefficient	-
ρ_ν	Spectral reflection coefficient	Hz^{-1}
σ_{jit}	Jitter standard deviation	arcsecond
σ_R	CCD readout noise	$\sqrt{\mathrm{e}^-}\,\mathrm{pixel}^{-1}$
τ	Transmission coefficient	-
τ_ν	Spectral transmission coefficient	Hz^{-1}
Φ	Radiant flux	W
Φ_ν	Spectral flux	$\mathrm{W\,Hz}^{-1}$
Ω	Solid angle	sr
Ω_A	Antenna solid angle	sr
Ω_{ap}	Aperture solid angle	sr
Ω_{mb}	Main beam solid angle	sr
Ω_{src}	Source solid angle	sr

1. Introduction

Astronomy, the study of celestial objects and phenomena, is largely dependent on the observation of electromagnetic radiation[1] emitted by the objects of interest. Unfortunately, wide ranges of the spectrum like the far-infrared (FIR) or ultraviolet (UV) range are partially or even completely blocked by atomic and molecular transitions within the atmosphere, e.g. the rotational transitions of the water molecule in the FIR. Only narrow spectral windows like the visible range are open and allow ground-based observations. "The answers to many fundamental, yet still unresolved astrophysical questions, such as those about the detailed mechanisms of astronomical engines, the secrets of exoplanet atmospheres, or the distribution of water in our own solar system, closely linked with questions about its own formation and evolution, thus lay obscured behind this atmospheric curtain"[1]. This circumstance is presented in the upper diagram of figure 1.1, which shows the atmospheric transmission at Mauna Kea in Hawaii in a wavelength range from 1 µm to 1000 µm.

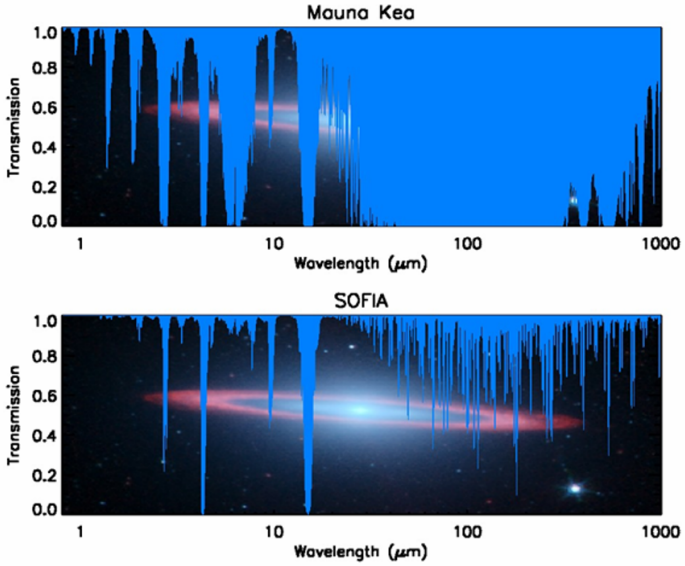


Figure 1.1.: The typical atmospheric transmission at a SOFIA observation altitude of 45,000 feet as compared to the transmission on a good night at Mauna Kea (13,800 ft. MSL)[2]. A clear improvement of the atmospheric transmission in the stratosphere can be observed for wavelengths larger than 1 µm.

The effects of the atmospheric extinction can be circumvented by moving to space-based observatories like Herschel Space Telescope, Hubble Space Telescope or James Webb Space Telescope. Space-based telescopes can observe the whole unobscured electromagnetic spectrum

[1] Besides the observation of electromagnetic radiation, there exist particle astronomy and gravitational wave astronomy which is emerging as a new branch of observational astronomy.

but have two major disadvantages: First of all, launching a telescope into space is a highly complicated and very expensive challenge. Due to the limitations on the payload size of rockets, large telescopes need sophisticated folding mechanisms. Secondly, once the space-based observatory is launched, it may, depending on the orbit, not be accessible for maintenance or hardware upgrades. Especially cryogenic coolant fluids (see the Herschel Space Observatory) or fuel cannot be refilled.
Airborne telescopes compromise between the observable spectral range and the maintainability as well as the costs of the mission. The lower diagram of figure 1.1 shows the atmospheric transmission at 13.7 km, a typical flight altitude for aircraft-based observatories which is much better compared to the atmospheric transmission on Mauna Kea. Aircraft-based observatories like Stratospheric Observatory for Infrared Astronomy (SOFIA) typically fly 10 – 12 hours allowing daily maintenance. However, the costs per flight are extremely high[2].
In contrast, stratospheric balloon observatories offer a flight duration in the order of weeks or even months, resulting in lower costs per observation hour. As stratospheric balloons can reach an altitude of up to 40 km, they allow observations with less perturbation as they leave over 99% of the atmospheric mass behind[3]. Furthermore larger telescopes with higher angular resolutions can be used for the observations as no fuselage is limiting the telescope size as compared to aircraft-based observatories. Nevertheless, they can be easily maintained and upgraded during the ground-time.

1.1. ESBO

After the retirement of Herschel due to a depletion of cryogenic coolant fluids, SOFIA is currently the only active far-infrared observatory with an expected lifetime of 20 years until 2030[2]. The planned European Stratospheric Balloon Observatory aims at developing a community-accessible stratospheric balloon observatory for the FIR band to extend the landscape of FIR observatories. The "concept focuses on reusable platforms with exchangeable instruments and telescopes performing regular flights"[1].
The design study ESBO-*DS* is continuing the project ORISON (innOvative Research Infrastructure based on Stratospheric balloONs) that studied the general feasibility of a balloon-based observatory and assessed the interests and scientific needs within the astronomical community. In addition to three flight systems, all necessary infrastructure to operate a stratospheric balloon observatory like proposal tools, ground systems and data pipelines are being conceptually developed.
To show the maturity of ESBO-*DS*, the prototype Stratospheric Ultraviolet Demonstrator of an Imaging Observatory (STUDIO) with a 0.5 m telescope will be built and launched in 2022. STUDIO features a UV imaging instrument with a field of view (FOV) of 30' x 30' for observations in the 200 nm to 300 nm wavelength regime. The development is motivated by two science cases: First of all, hot compact stars can be easily detected in the UV band because their emitted flux is increasing towards the UV band. In contrast, the flux of other stars is decreasing because of their lower temperature. The observation of hot compact stars like white dwarfs is of great importance for astroseismic models. Binary systems of white dwarfs emit strong gravitational waves and can be used as calibration sources by future gravitational wave observatories like eLISA.

[2]SOFIA costs about 1,000,000 $ per flight.

Secondly, ESBO-*DS* allows the continued monitoring of the strong emission line of ionized magnesium (Mg II) during flares of red dwarf stars. These flares result from reconfigurations of the stellar magnetic field and play an important role in the evolution of planets and other circumstellar matter. Systematic multi-wavelength monitoring of flare stars and the determination of the flare's occurrence rate promises new insights on the physics of these outbursts[1].
By studying concepts for larger following platforms, ESBO-*DS* shall pave the way towards European Stratospheric Balloon Observatory (ESBO). Following STUDIO, ESBO shall provide a 1.5 m NIR telescope and a 5 m FIR telescope platform in the long term. All platforms feature exchangeable science instruments for community-based access. Further down the road, the platforms shall be opened to allow the development and usage of principle investigator (PI) instruments.
The project ESBO-*DS* is financed by the European Union's Horizon 2020 Programme and carried out by a consortium consisting of the Institute of Space Systems at the University of Stuttgart, the Swedish Space Corporation, the Institute for Astronomy and Astrophysics at the University of Tübingen, the Max Planck Institute for extraterrestrial Physics and the Instituto de Astrofísica de Andalucía[4].

1.2. Exposure Time Calculator & Requirements

During the development of ESBO-*DS* and ESBO, different telescope designs, materials and observation strategies are under consideration. The decision for a specific component requires an estimate of the future observatory's performance given the component's properties. A modular end-to-end simulation tool of an observatory can standardize and significantly accelerate this decision process.
Once a telescope design is selected and built, the European Stratospheric Balloon Observatory may start its operational life. Astronomers are then allowed to submit proposals for observations. These proposals must include, inter alia, a detailed explanation of the scientific gain as well as an estimate for the necessary observation time. To assess the necessary observation time, a tool to simulate the observation of the specified astronomical target and observation conditions using the ESBO platform is needed.
In a later stage of ESBO's operational lifetime, PI-instruments will be used besides the facility instruments. Again, a simulation tool is required to enable the developers of PI-instruments to investigate the performance of their detector concepts for a fixed telescope design.
This said the requirements on a new simulation tool have been collected and documented in a Software Requirements Specification[5]. The most important requirements are

- **Configurability** to allow flexible modeling of complex science cases or telescope setups.
- **Modularity** by object-oriented programming to enable the integration of future extensions.
- **Library** of components including:
  - **Targets**: black body target, target from a file of spectral flux densities
  - **Optical components**: cosmic background (black body), arbitrary stray light, atmosphere, mirror, lens, filter, beam splitter

  - **Detector**: imaging detector, heterodyne detector

- **Disturbances** like the telescope's point spread function (PSF) or pointing jitter shall be considered

An exposure time calculator (ETC), a software tool used to calculate the required exposure time to reach a predefined signal-to-noise ratio (SNR) with an observatory given an astronomical target and observation conditions, can satisfy all of the aforementioned use cases and requirements. In detail, the calculations of an ETC start by modeling the emission of electromagnetic radiation by an astronomical target as well as possible background emission. Subsequently, the transfer of the emitted radiation through several optical components like galactic dust clouds or atmospheric components to the telescope is simulated. Of course, the telescope including all optical components like mirrors, lenses or filters is modeled as well to improve the precision of the simulation. Finally, the detection process and the corresponding statistical effects are simulated resulting in either a value for the necessary exposure time, a value for the reached SNR or a value for the limiting sensitivity of the telescope. This process is also presented in figure 1.2.

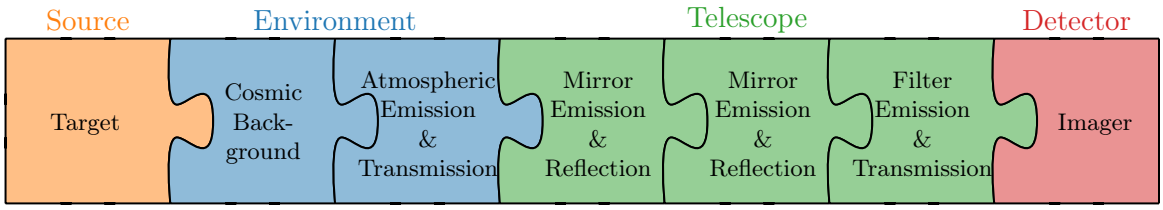


Figure 1.2.: This illustration shows the concept for a modular exposure time calculator. The software provides a library of different components that can be arranged to model the radiation's path from the source to the detector.

Scientific observations require a high SNR which shall be reached in a preferably short exposure time. This said an exposure time calculator is the right tool to estimate the performance of a specific observation setup. Therefore a highly modular exposure time calculator (in the following called ESBO-ETC) was developed during this master thesis.

1.3. Thesis Overview

This master thesis is divided into several chapters, starting with some important physical basics on the transfer and detection of radiation in chapter one. This is followed by a chapter on the necessary preparatory work to implement the code for the exposure time calculator. The fourth chapter describes the software architecture, numerical approaches, input and output of the software as well as the documentation in detail. Chapter five continues with the verification of the software using other approved ETCs and some hand calculations. Finally, the ETC is applied to two science cases to validate the software. First, the influence of different mirror coatings on the integration time is assessed for SOFIA/FORECAST. Afterward, the performance of the future ESBO telescope concept is investigated for a line observation with a heterodyne instrument. The thesis is closed by a final discussion of the results of the work and an outlook on future improvements.

2. Theory

An exposure time calculator models the emission, transfer and detection of radiation from astronomical objects of interest. The development of an ETC requires a sound understanding of the physical properties of electromagnetic (EM) radiation, its transfer & detection mechanisms and the statistics of measured signals. These prerequisites are explained in this chapter, starting with some fundamental units of the EM radiation.

2.1. Fundamentals

The energy of EM radiation is carried by photons, which follow the laws of quantum mechanics. According to the wave-particle duality of quantum mechanics, photons can be described either as a wave or as a particle. On the one hand, EM radiation can be modeled as perpendicular electric and magnetic waves propagating with the speed of light c through vacuum. The derivations for the heterodyne instrument described in section 2.4.2 make use of the light's wave nature. On the other hand, electromagnetic radiation can be described as particle radiation with photons carrying the radiation energy. This theory is used by the derivations for the imaging detector in section 2.4.1.
EM radiation is emitted by various processes in nature with the thermal emission being the most prominent one. Every particle or body with a temperature T above 0 K emits a continuous spectrum of electromagnetic radiation following Planck's law.

$$L_{\Omega,\nu}(T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{\frac{h\nu}{k_B T}} - 1} \tag{2.1}$$

with the Planck constant h, the Boltzmann constant k_B and the radiation frequency ν.
Another important process is the emission (and absorption) of discrete EM radiation frequencies (lines) by the transition of quantum states in atoms and molecules. Due to the uncertainty principle of state lifetime and the thermal velocity distribution, these discrete lines reveal a natural line width. Each atom and molecule is uniquely identified by its great abundance of different spectral lines allowing astronomers to deduce the composition of distant objects by analyzing their spectrum. This mechanism is relevant for both the emission of the astronomical target as well as for the extinction in the atmosphere for example.
Depending on the problem, several radiometric quantities can be used to characterize sources and receivers of EM radiation. Each quantity can be defined for a radiation source denoted by the term *radiant* or for a receiver indicated by the term *irradiant*. Furthermore, all following physical quantities can be defined as spectral quantities analogous by differentiating with respect to the unit wavelength $\partial\lambda$ or the unit frequency $\partial\nu$. An index x_λ or x_ν denotes such a spectral quantity. They can be converted using the identities

$$c = \lambda \cdot \nu \tag{2.2}$$

$$x_\lambda = x_\nu \cdot \frac{c}{\lambda^2} = x_\nu \cdot \frac{\nu^2}{c}.$$

which arise by equating the two differentials $x_\nu = \frac{\mathrm{d}x}{\mathrm{d}\nu}$ and $x_\lambda = \frac{\mathrm{d}x}{\mathrm{d}\lambda}$.

Radiant Flux

First of all, sources and receivers of EM radiation can be characterized by the energy ∂Q transported per time unit ∂t which gives the radiant flux Φ

$$\Phi = \frac{\partial Q}{\partial t} \tag{2.3}$$

in units of W. The spectral flux is defined analogous as

$$\Phi_\nu = \frac{\partial^2 Q}{\partial t \, \partial \nu}$$

in units of $\frac{\mathrm{W}}{\mathrm{Hz}}$. The radiant flux, as well as the spectral flux, describe the total emitted or received radiation power

Radiant flux density

The radiant flux density E is defined as the radiant flux per unit area ∂A.

$$E = \frac{\partial \Phi}{\partial A} \tag{2.4}$$

$$E_\nu = \frac{\partial^2 \Phi}{\partial A \, \partial \nu}$$

The units of the (spectral) radiosity are $\frac{\mathrm{W}}{\mathrm{m}^2}$ and $\frac{\mathrm{W}}{\mathrm{m}^2 \cdot \mathrm{Hz}}$ respectively. This quantity can be used to measure the power received by a detector per surface area.

Radiance

The radiance or brightness L_Ω is a measure of the spatial and directional strength of a source. It is defined as the radiant flux $\partial \Phi$ emitted or received by any surface per solid angle $\partial \Omega$ and per projected surface area $\partial A \cos(\theta)$ according to the cosine-law. The index x_Ω indicates a directional quantity.

$$L_\Omega = \frac{\partial^2 \Phi}{\partial \Omega \, \partial A \cos(\theta)} \tag{2.5}$$

$$L_{\Omega,\nu} = \frac{\partial^3 \Phi}{\partial \Omega \, \partial A \cos(\theta) \, \partial \nu}$$

The units of the (spectral) radiance are given as $\frac{\mathrm{W}}{\mathrm{sr} \cdot \mathrm{m}^2}$ and $\frac{\mathrm{W}}{\mathrm{sr} \cdot \mathrm{m}^2 \cdot \mathrm{Hz}}$.

Brightness Temperature

The brightness temperature T_b is usually used by radio astronomers to describe the brightness of sources. As radio astronomers observe in the lower frequency domain, the Rayleigh-Jeans approximation of Planck's law holds for $h\nu \ll k_B T$, yielding

$$T_B = L_{\Omega,\nu} \cdot \frac{c^2}{2 \cdot k_B \cdot \nu^2} = L_{\Omega,\nu} \cdot \frac{\lambda^2}{2 \cdot k_B} = L_{\Omega,\lambda} \cdot \frac{\lambda^4}{2 \cdot k_B \cdot c} \tag{2.6}$$

using relation (2.2).

Noise Temperature

The noise temperature of a component is used in electronics and radio technology to describe the level of Johnson-Nyquist noise power introduced by this component. In the Rayleigh-Jeans approximation, the power spectral density (PSD) P_ν of a signal can be expressed as the temperature T of a resistor with an equal PSD

$$\frac{P}{\Delta\nu} = P_\nu = k_B T \tag{2.7}$$

with the noise bandwidth $\Delta\nu$[6] over which the noise power is measured.

2.2. Radiation Sources

Two limiting cases exist for the spatial extend of radiation sources: an infinitesimally small point source and an extended source which covers at least the full FOV of the telescope (in the following the term *extended source* refers to this limiting case). Each case allows the measurement of a different radiation quantity. "When the emitting region is larger than the beam the brightness of the region covered by the beam can be measured. [...] However, when the source is much smaller than the telescope beam its solid angle cannot be determined – we say that the source is unresolved. As a result we cannot measure its brightness and only the more limited information, the flux density [...], is available"[6]. Consequently, the spectral flux density E_ν must be used to model point sources and the radiance $L_{\Omega,\nu}$ for largely extended sources. Both cases are considered by ESBO-ETC

Black Body

Many astronomical sources resemble a black body radiator with a spectral distribution according to Planck's law (2.1). The brightness of these sources can be described using the apparent magnitude system. In agreement to the human eye, the magnitude system uses a logarithmic scale with the brightness of Vega defined as $0.03\ ^{\text{mag}}$ as reference. The apparent magnitude of any object is defined as

$$m = -2.5 \cdot \log_{10}\left(\frac{E}{E_{Vega}}\right). \tag{2.8}$$

To model a black body radiator with a given apparent magnitude, Planck's law from equation (2.1) has to be normalized and scaled to tabulated values of the flux density of a $0\ ^{\text{mag}}$ star as shown in table A.1. The spectral band used for this process can be selected based

on the wavelength range of interest. Afterward, the spectral quantity is scaled to the desired apparent magnitude using equation (2.8).
In the case of a point source, the object's spectral flux density for a given black body temperature T, apparent magnitude m and central band wavelength λ_0 can be calculated as

$$E_\nu = E_{\nu,Planck}\left(T\right) \cdot \frac{E_{\lambda,Vega}\left(\lambda_0\right)}{E_{\lambda,Planck}\left(\lambda_0, T\right)} \cdot 10^{-\frac{2}{5}\cdot m}. \tag{2.9}$$

In the other case of an extended source, the objects surface brightness S is used for the calculation of the black body's radiance

$$L_{\Omega,\nu} = L_{\Omega,\nu,Planck}\left(T\right) \cdot \frac{E_{\lambda,Vega}\left(\lambda_0\right)}{L_{\Omega\lambda,Planck}\left(\lambda_0, T\right) \cdot 1\,\mathrm{sr}} \cdot 10^{-\frac{2}{5}\cdot \mathrm{S}\cdot 1\,\mathrm{sr}}. \tag{2.10}$$

2.3. Radiative Transfer

An astronomical object emits EM radiation that travels along the line of sight to the telescope where the radiation is focused by mirrors and lenses onto the focal plane. This radiation is called the signal radiation or just the signal. The signal passes on this path several different optical components with different optical properties which may act on the signal as indicated by the upper flow in figure 2.1. On the one hand, the signal might be attenuated by dust, gases like the atmosphere, mirrors, lenses, etc. Mathematically, a spectral transmittance τ_ν, reflectance ρ_ν or absorption α_ν coefficient is assigned to each of the aforementioned optical components with $0 \leq \tau_\nu,\ \rho_\nu,\ \alpha_\nu \leq 1$. The radiance $L_{\Omega,\nu}{}'$ after the component is then given as

$$L_{\Omega,\nu}{}' = \tau_\nu \cdot L_{\Omega,\nu} \tag{2.11}$$

with the radiance $L_{\Omega,\nu}$ before the component. This relation holds also for reflection ρ_ν and the complement of the absorption $1 - \alpha_\nu$.
On the other hand, the radiation received by the telescope might be enhanced by other radiation sources like the cosmic background, galactic cirrus, atmospheric emission or thermal emission of the telescope components. All these sources sum up to give the background radiation in contrast to the signal radiation as illustrated in figure 2.1. The background radiation can be subtracted from the received radiation by observation techniques like chopping or nodding to obtain only the signal radiation[7]. However, it is impossible to remove the noise of the background radiation as shown in the calculation of the SNR in section 2.4.
Another disturbance mechanism is the obstruction of the telescope's FOV by structural or optical components like an opposing secondary mirror. The fraction of the aperture obstruction o can be calculated as the ratio of the obstructing component's area A_{obs} and the aperture area A_{ap}

$$o = \frac{A_{obs}}{A_{ap}} \tag{2.12}$$

which follows $0 \leq o \leq 1$. The radiance $L_{\Omega,\nu}{}'$ after the obstructing component can be calculated by

$$L_{\Omega,\nu}{}' = o \cdot L_{\Omega,\nu}. \tag{2.13}$$

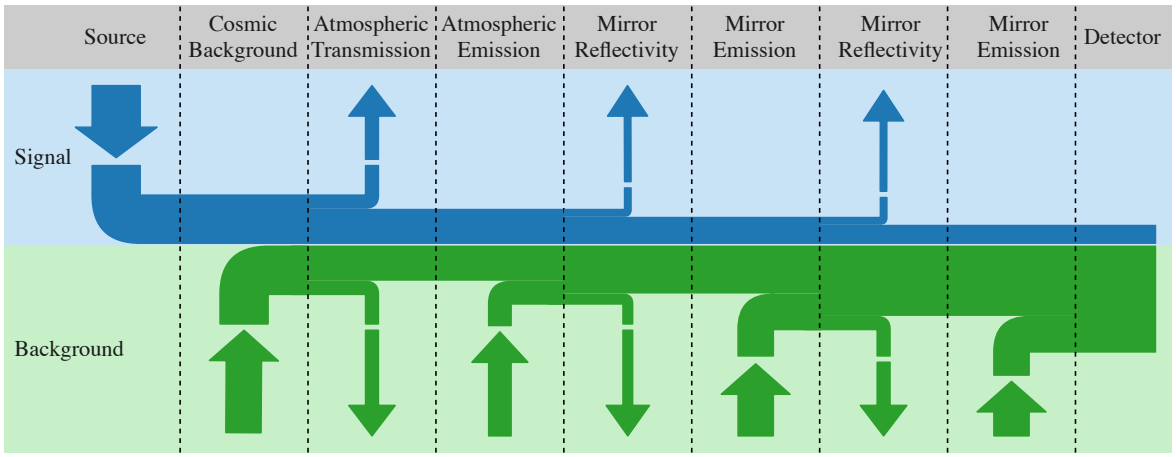


Figure 2.1.: Contributions to the signal and background radiation
The signal decreases on its way from the source to the detector while the background radiation increases due to the thermal emissions of the cosmic background, the atmosphere and the telescope's components.

The telescope's PSF introduces another distortion on the received radiation. Diffraction coupled with aberrations in the optical components "will cause the image of a perfect point to be smeared out into a blur spot occupying a finite area of the image plane"[8]. This said the PSF describes the image of a perfect point source on the focal plane of the telescope. The point spread function of an arbitrary telescope without taking into account optical aberrations can be calculated as the Fourier transform of its aperture with the Airy disk being the simplest case for a circular aperture. The airy disk intensity function $I_\lambda(\theta)$ is given as

$$I_\lambda(\theta) = I_0 \cdot \left(\frac{2J_1(x)}{x}\right)^2 \tag{2.14}$$

with the peak intensity I_0, the angle of observation θ, the aperture diameter D_{ap} and the substitution $x = \frac{2\pi}{\lambda}\frac{D_{ap}}{2}\sin(\theta) \approx \frac{2\pi}{\lambda}\frac{D_{ap}}{2}\theta$. The angular resolution of a telescope can be determined with the full width half maximum (FWHM) of its PSF.

Obstructing components like an opposing secondary mirror not only reduce the received radiation but also alter the telescope's aperture and therefore the telescope's PSF. The airy disk with a central circular obstruction can be described by

$$I_\lambda(\theta) = \frac{I_0}{(1-o)^2} \cdot \left(\frac{2J_1(x)}{x} - \frac{2\sqrt{o}J_1(\sqrt{o}x)}{x}\right)^2. \tag{2.15}$$

As a result, more energy is diffracted from the central disc into the outer rings as shown in figure 2.2, leading to a higher FWHM and therefore to a lower angular resolution of the telescope. The position of the FWHM of the unobstructed airy disk is given by $\theta = 0.514 \cdot \frac{\lambda}{D_{ap}}$ whereas the position of the FWHM cannot be analytically calculated in the obstructed case.

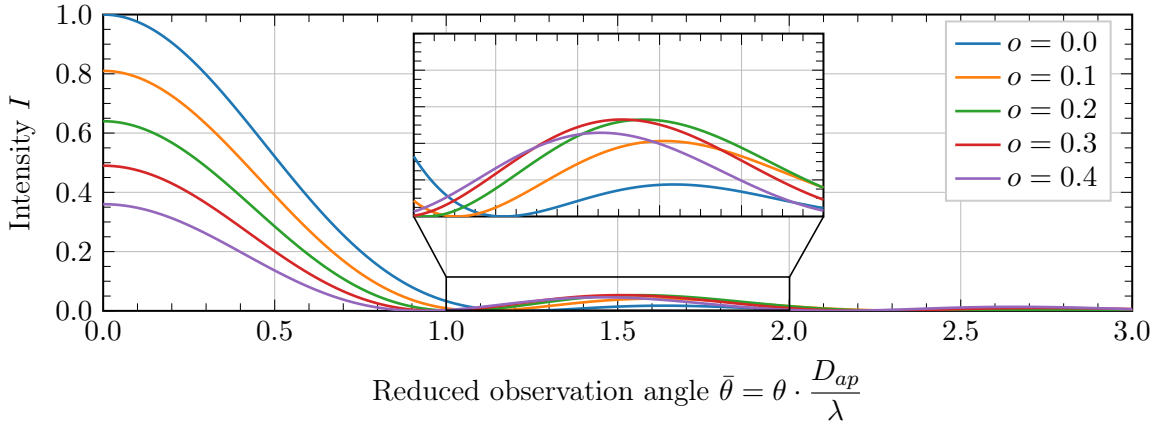


Figure 2.2.: Airy disc for different obstruction ratios o
More energy is diffracted from the central disc into the outer rings with an increasing obstruction ratio.

2.4. Radiation Detection

After both the signal and the background radiation reach the focal plane, the electromagnetic radiation must be measured. Today, astronomers use semiconducting detectors to convert the incident radiation into an electrical signal which can be further processed by computers. Two types of detectors are available in the first version of ESBO-ETC: an imaging detector which detects the spatial distribution of the radiation and a heterodyne detector to detect the spectral distribution. Both detector types, their basic principles and the calculation of the SNR from the received signal and background radiation are explained in the following.

2.4.1. Imaging Detector

An Imaging detector measures the spatial distribution of the EM radiation. Hence the detector integrates over the whole detectable wavelength regime, the measurement is spectrally not resolved. Typical examples for imaging detectors are CCD sensors and the more modern but in astronomy less prominent CMOS sensors.
An imaging detector consists of an array of millions of pixels. The size of these pixels defines the spatial resolution of the sensor. Each pixel is an independent photodetector that converts the incident radiation into an electrical signal. In the case of the prominent CCD sensor, each pixel is a semiconductor in which incident photons create free electrons. The ability of photons to create a free electron in the pixel is dependent on the photon's wavelength and can be quantified by the pixel's spectral quantum efficiency QE_ν.
During the exposure of the pixel, lots of electrons are freed and collected in the pixel's well. Due to the temperature of the sensor, additional free thermal electrons are created and contained in the well. These electrons are called the dark current I_D of the pixel and can only be reduced by lowering the detector's temperature. By taking dark frame images with no incident radiation, this dark current background can be removed from the science image. However, as shown later, the noise of the dark current cannot be removed from the science images.

To measure the number of collected electrons, the charge packets are moved by a clock to the transfer registers as shown in figure 2.3. These registers move the collected electrons to an analog to digital converter (ADC) which converts the collected charge to a digital signal. This conversion introduces additional noise, known as the readout noise σ_R per pixel. The readout of the pixels can be done in different ways, either after the exposure of the pixels or by continuously measuring the charge, which is commonly known as following up the ramp.

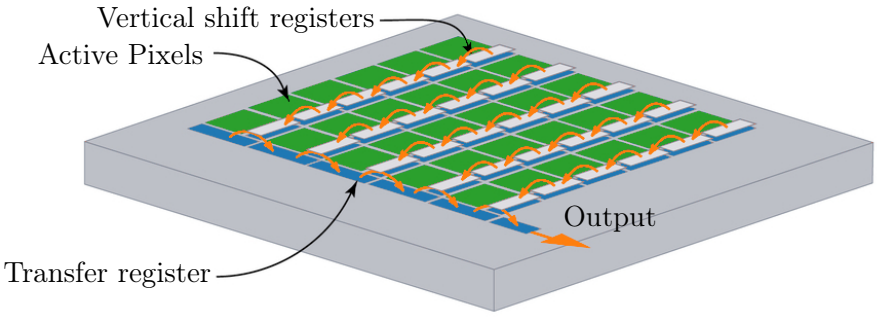


Figure 2.3.: Example read out of a CCD sensor
The charge packets are transferred over the neighboring pixels to a transfer register. The output of the transfer register is connected to an ADC.
Source: `https://www.stemmer-imaging.com/en-pl/knowledge-base/ccd/`

Each pixel well has a finite well capacity that overflows to the neighboring pixels if the pixel is overexposed. This effect is called blooming and must be avoided as this irreversibly alters the measurement.

Exposure

Before the SNR can be calculated, the number of incident photons per pixel has to be computed from the spectral flux of the incident radiation. Multiplying the spectral flux Φ_ν with the exposure time t_{exp} and dividing by the photon energy $h\nu$ yields the spectral photon count.

$$N_{phot,\nu} = \frac{\Phi_\nu \cdot t_{exp}}{h \cdot \nu} \tag{2.16}$$

The number of produced free electrons can then be calculated using the pixel's spectral quantum efficiency QE_ν as

$$N_{e^-} = \int N_{phot,\nu} \cdot QE_\nu \, \mathrm{d}\nu = \int \frac{\Phi_\nu \cdot t_{exp}}{h \cdot \nu} \cdot QE_\nu \, \mathrm{d}\nu \tag{2.17}$$

The aforementioned spectral flux of the incident radiation at the focal plane position of a pixel depends both on the shape of the source as well as on the PSF of the telescope. As explained in section 2.2, the spectral radiance is used to describe extended sources whereas only the spectral radiant flux density can be measured for point sources.

Point Source with Unobstructed Aperture In the simple case of a point source with an unobstructed aperture, the spectral flux can be calculated as

$$\Phi_\nu = E_\nu \cdot A_{ap} \tag{2.18}$$

Obstructed Point Source with Obstructed Aperture The case of a point source with an obstructed aperture is similar to the unobstructed case except for the correction factor $(1 - o)$.

$$\Phi_\nu = E_\nu \cdot A_{ap} \cdot (1 - o) \tag{2.19}$$

Extended Source with Unobstructed Aperture In the case of an extended source (the background radiation can also be treated as an extended source), the incident spectral flux Φ_ν at the focal plane can be calculated with the viewing factor from the extended source to the detector pixel. As the extended source covers the whole aperture, the calculation can be reduced to the viewing factor from the aperture to the detector pixel as illustrated in figure 2.4.
According to [9], the radiant flux $\Phi_{\nu,1-2}$ from the aperture A_1 to a quadratic detector pixel A_2 with the edge length l can be calculated as

$$\Phi_{\nu,1-2} = \iint L_{\Omega,\nu,1} \cdot \frac{\cos^2(\beta)}{r^2} \, \mathrm{d}A_1 \, \mathrm{d}A_2 \tag{2.20}$$

with the line of sight r and the angle β from figure 2.4a. The two sketches 2.4a and 2.4b show that r can be represented using the Pythagorean theorem and the trigonometric functions as

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + f^2} = \sqrt{(R \cdot \cos(\varphi) - x_2)^2 + (R \cdot \sin(\varphi) - y_2)^2 + f^2}. \tag{2.21}$$

Applying the cosine-function on the tilt angle β of the line of sight in figure 2.4a and using equation (2.21) results in

$$cos(\beta) = \frac{F}{r} = \frac{F}{\sqrt{(R \cdot \cos(\varphi) - x_2)^2 + (R \cdot \sin(\varphi) - y_2)^2 + f^2}}. \tag{2.22}$$

The infinitesimal area element $\mathrm{d}A_1$ can be rewritten in polar coordinates using figure 2.4c. For an infinitesimal angle $\mathrm{d}\varphi$, the edges of $\mathrm{d}A_1$ become orthogonal yielding

$$\mathrm{d}A_1 = \mathrm{d}R \cdot R \cdot \sin(\mathrm{d}\varphi) = R\mathrm{d}R\mathrm{d}\varphi \tag{2.23}$$

which can be simplified for infinitesimal small angles $\mathrm{d}\varphi$. Plugging equation (2.21), (2.22) and (2.23) into equation (2.20) and applying the integration bounds gives

$$\Phi_{\nu,1-2} = L_{\Omega,\nu,1} \cdot \int_{-\frac{l}{2}}^{\frac{l}{2}} \int_{-\frac{l}{2}}^{\frac{l}{2}} \int_{0}^{\frac{D_{ap}}{2}} \int_{0}^{2\pi} \frac{F^2}{\left((R \cdot \cos(\varphi) - x_2)^2 + (R \cdot \sin(\varphi) - y_2)^2 + f^2\right)^2} \cdot R \, \mathrm{d}\varphi \, \mathrm{d}R \, \mathrm{d}x_2 \, \mathrm{d}y_2. \tag{2.24}$$

The sensor pixel edge length l is usually much smaller than the telescope's aperture diameter D_{ap}: $x_1 = R \cdot \cos(\varphi) \gg x_2$ as well as $y_1 = R \cdot \sin(\varphi) \gg y_2$. Therefore, eq. (2.24) can be

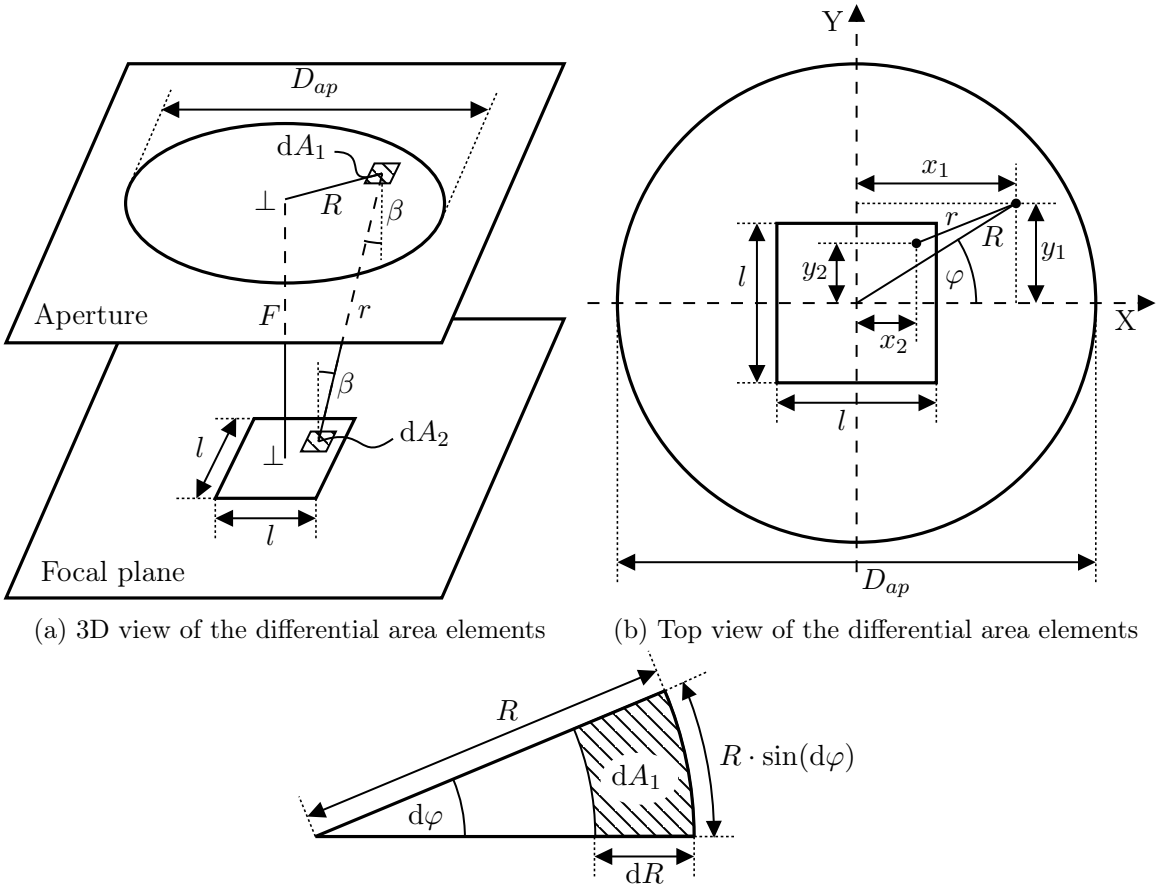
simplified to

$$\Phi_{\nu,1-2} = L_{\Omega,\nu,1} \cdot \int_{-\frac{l}{2}}^{\frac{l}{2}} \int_{-\frac{l}{2}}^{\frac{l}{2}} \int_{0}^{\frac{D_{ap}}{2}} \int_{0}^{2\pi} \frac{F^2}{\left(\left(R \cdot \cos(\varphi)\right)^2 + \left(R \cdot \sin(\varphi)\right)^2 + f^2 \right)^2} \cdot R \, \mathrm{d}\varphi \, \mathrm{d}R \, \mathrm{d}x_2 \, \mathrm{d}y_2. \tag{2.25}$$

Carrying out the integrals and simplifying using the definition of the focal number $N = \frac{f}{D_{ap}}$ leads to the final equation for the flux received by a detector pixel from an extended source

$$\boxed{\Phi_{\nu,1-2} = L_{\Omega,\nu,1} \cdot l^2 \cdot \frac{\pi}{4N^2 + 1}}. \tag{2.26}$$

As this equation is equivalent to the equation used by the community approved exoplanet transit simulator *ExoSim*, the approximations above are valid. The detailed derivation of the equation used by ExoSim is given in appendix B.



(a) 3D view of the differential area elements

(b) Top view of the differential area elements

(c) Differential area in a circle

Figure 2.4.: Sketch for the calculation of the viewing factor from the aperture to a sensor pixel

Extended Source with Obstructed Aperture The derivation of the radiant flux received by a sensor pixel from an extended source with an obstructed aperture is analog to the derivation in

the unobstructed case if a central, circular obstruction is assumed. This assumption holds for most telescope designs, as the obstruction is mostly caused by an opposing secondary mirror in the center of the aperture. However, the integration with respect to R in equation (2.25) behaves differently due to a dependency of $L_{\Omega,\nu,1}$ on R. In detail, the radiance of the central obstructing component is different than the radiance of the extended source as the obstructing component may emit its own thermal radiation.

$$L_{\Omega,\nu,1}(R) = \begin{cases} L_{\Omega,\nu,1,obs} & \text{for } R \leq \sqrt{o} \cdot \frac{D_{ap}}{2}, \\ L_{\Omega,\nu,1} & \text{otherwise} \end{cases} \tag{2.27}$$

After executing the integrations with respect to φ, x_2 and y_2 in eq. (2.25), the received radiant flux can be expressed as

$$\Phi_{\nu,1-2} = 2\pi \cdot l^2 \cdot \int_0^{\frac{D_{ap}}{2}} L_{\Omega,\nu,1}(R) \cdot \frac{f^2 \cdot R}{\left(R^2 + f^2\right)^2} \, \mathrm{d}R. \tag{2.28}$$

Plugging equation (2.27) into (2.28) and executing the integration leads to

$$\begin{aligned} \Phi_{\nu,1-2} = 2\pi \cdot l^2 \cdot \Bigg(& L_{\Omega,\nu,1,obs} \left[\frac{1}{2} - \frac{f^2}{2\left(f^2 + \frac{1}{4} \cdot o \cdot {D_{ap}}^2\right)} \right] + \\ & L_{\Omega,\nu,1} \left[\frac{f^2}{2\left(f^2 + \frac{1}{4} \cdot o \cdot {D_{ap}}^2\right)} - \frac{f^2}{2\left(f^2 + \frac{1}{4} \cdot {D_{ap}}^2\right)} \right] \Bigg) \end{aligned} \tag{2.29}$$

which can be simplified to

$$\Phi_{\nu,1-2} = \pi \cdot l^2 \cdot \left(L_{\Omega,\nu,1,obs} \cdot \frac{o}{4N^2 + o} + L_{\Omega,\nu,1} \cdot \frac{1 - o}{4N^2 + 1 + o \cdot \left(1 + \frac{1}{4N^2}\right)} \right). \tag{2.30}$$

For $\frac{1}{4N^2} \ll 1$ and $4N^2 \gg 1 \geq o$ which holds for most telescopes, the flux received by an obstructed detector pixel can be calculated by

$$\boxed{\Phi_{\nu,1-2} \approx l^2 \cdot \frac{\pi}{4N^2 + 1} \cdot \left(L_{\Omega,\nu,1,obs} \cdot o + L_{\Omega,\nu,1} \cdot (1 - o)\right)} \tag{2.31}$$

which exhibits a great similarity to eq. (2.26) in the unobstructed case by the factor $l^2 \cdot \frac{\pi}{4N^2+1}$. It follows, that the radiances of the extended source and the obstructing component can be approximately summed after weighting with the obstruction factor o. The error of used approximations is assessed in appendix C.

Spatial Distribution on Pixels As explained in section 2.3, the telescope's PSF spreads the image of a perfect point across the focal plane. However, this only comes into effect for the case of a point source. In the case of an extended source, the image consists of an infinite amount of blurred points by the PSF, summing up to a uniformly exposed image without any visible impact by the PSF.

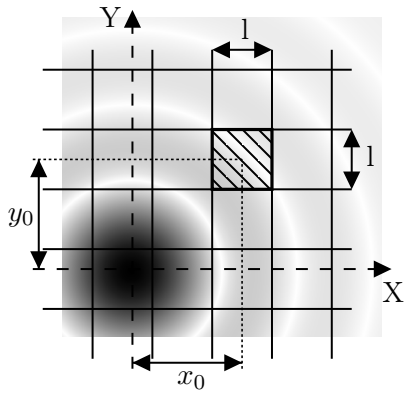


Figure 2.5.: Mapping the PSF onto the pixel grid
An exemplary PSF is shown on a pixel grid. A single pixel is highlighted and the distance to the PSF's center is labeled.

As shown in figure 2.5, the incident spectral radiant flux per pixel depends on both the size and the position of the pixel relative to the center of the PSF. Mathematically, $\Phi_{\nu,pix}$ can be calculated as

$$\Phi_{\nu,pix} = \Phi_\nu \cdot \frac{\int_{x_0-\frac{l}{2}}^{x_0+\frac{l}{2}} \int_{y_0-\frac{l}{2}}^{y_0+\frac{l}{2}} I\left(x,y\right)\,\mathrm{d}x\,\mathrm{d}y}{\iint_{\mathbb{R}^2} I\left(x,y\right)\,\mathrm{d}x\,\mathrm{d}y} \tag{2.32}$$

with the intensity function of the PSF $I\left(x,y\right)$ and the pixel center offset (x_0, y_0). In the developed ETC, these integrals are numerically computed as explained in section 4.2.2. The

pointing of a telescope often suffers from pointing jitter due to mechanical vibrations or oscillations of the controllers. In the simplest case, the jitter has zero mean and follows the Gaussian normal distribution[10]

$$f\left(x,y,\sigma_{jit}\right) = \frac{1}{2\pi{\sigma_{jit}}^2} \cdot e^{-\frac{x^2+y^2}{2{\sigma_{jit}}^2}} \tag{2.33}$$

with its standard deviation σ_{jit}. Pointing jitter acts on the image in the same way as the PSF by smearing out the image of a perfect point source. Hence the influence of the pointing jitter on the incident spectral radiant flux per pixel can be calculated as the convolution of the jitter and the PSF function

$$\Phi_{\nu,pix} = \Phi_\nu \cdot \frac{\int_{x_0-\frac{l}{2}}^{x_0+\frac{l}{2}} \int_{y_0-\frac{l}{2}}^{y_0+\frac{l}{2}} I\left(x,y\right) * f\left(x,y,\sigma_{jit}\right)\,\mathrm{d}x\,\mathrm{d}y}{\iint_{\mathbb{R}^2} I\left(x,y\right) * f\left(x,y,\sigma_{jit}\right)\,\mathrm{d}x\,\mathrm{d}y}. \tag{2.34}$$

Photometric Aperture

Usually, the observation's SNR is not calculated for a single pixel but rather for a virtual photometric aperture (in the following only photometric aperture) containing a specific percentage of the incident radiation energy. This approach is necessary as typically many pixels are exposed during the observation due to the radiation's spatial distribution as explained in

the previous section 2.4.1. The total collected radiation in this photometric aperture is used for the calculation of the signal-to-noise ratio.
According to the defined requirements[5] ESBO-ETC shall implement both a circular and a quadratic photometric aperture. The radius and the edge length respectively of the photometric aperture can be either defined in the configuration file or calculated from a specified percentage of encircled energy EE. In the latter case, the equation

$$EE = \frac{\int_0^{\frac{D_{phot}}{2}} \int_0^{2\pi} I\left(r,\varphi\right) * f\left(r,\varphi,\sigma_{jit}\right)\ \mathrm{d}\varphi\,\mathrm{d}r}{\int_0^{\infty} \int_0^{2\pi} I\left(r,\varphi\right) * f\left(r,\varphi,\sigma_{jit}\right)\ \mathrm{d}\varphi\,\mathrm{d}r} \tag{2.35}$$

has to be solved for the diameter D_{phot} of the photometric aperture. This diameter is used for both the circular aperture as well as for the edge length of the quadratic aperture.
Besides providing a percentage of encircled energy, the keywords *peak*, *fwhm*, *min* can be supplied for a single pixel aperture, an aperture up to the PSF's FWHM and an aperture to the first minimum of the PSF. More information on this proceeding is given in section 4.2.2.

SNR Calculation

Photon counting devices like the CCD detector make use of the particle nature of the EM radiation. The statistics of arriving photons at the detector can be described with the Poisson distribution. For a large number N of photons, this distribution tends towards the Gaussian distribution with the standard deviation

$$\sigma = \sqrt{N_{phot}} \tag{2.36}$$

which holds for most astronomical observations[11]. Therefore the photon noise can be calculated as $\sqrt{N_{phot}}$.
Even though the bias of the background with contributions from the astronomical & thermal background radiation, dark noise and read-out noise can be removed from the image by observation techniques like chopping, nodding and dark frames, their noise contribution cannot be removed. This circumstance is considered in the *CCD-equation* developed by Howell[12]

$$\mathrm{SNR} = \frac{N_{e^-,sig}}{\sqrt{N_{e^-,sig} + N_{e^-,bg} + N_{pix} \cdot I_D \cdot t_{exp} + \left(N_{pix} \cdot \sigma_R\right)^2}} \tag{2.37}$$

with the dark current I_D and the readout noise σ_R. This equation is the central equation for the imaging detector, allowing to calculate both the SNR as well as the necessary exposure time from the electron counts.

2.4.2. Heterodyne Spectrometer

Unlike the imaging detector, a spectrometer is used to measure the spectral and not the spatial distribution of the incident radiation. However, the frequencies of the infrared bands (430 THz – 300 GHz) are too high for computerized signal processing and most electronic components. For this reason, the heterodyne principle is used to reduce the signal frequencies to a level that can be further processed by digital electronics.
Heterodyning describes the process of mixing two signals, creating two frequency-shifted signals with one signal at the sum and the other at the difference of the input frequencies. Typically, one of the two signals comes from a collector like a telescope and the other one from a local oscillator (LO) which can be adjusted in a certain range. The basic principle of a heterodyne receiver originates from radio technology where it is used to separate the information from the carrier signal.
Figure 2.6 shows a simple heterodyne receiver, where one signal is collected by a telescope and the second signal originates from an LO. A diplexer, which typically consists of one or more beamsplitters, combines the two signals which are subsequently fed into the non-linear mixer. After the two signals have been mixed, the desired intermediate frequency (IF) is filtered out and amplified by a preamplifier This amplifier is the first in a chain of amplifiers and must therefore only add as little noise as possible to the signal as this noise will be amplified by the amplifiers further down the signal path. Hence the preamplifier is kept as cool as possible to reduce the noise to a minimum. The overall noise power introduced by the receiver is described by the receiver noise temperature T_{rec} (see equation (2.7) which is typically in the order of 10^3 K).
After the IF signal has been sufficiently amplified, the spectrum is measured and digitized by a spectrometer in the back end. The efficiency of the back end is quantified by the back end degradation factor κ.

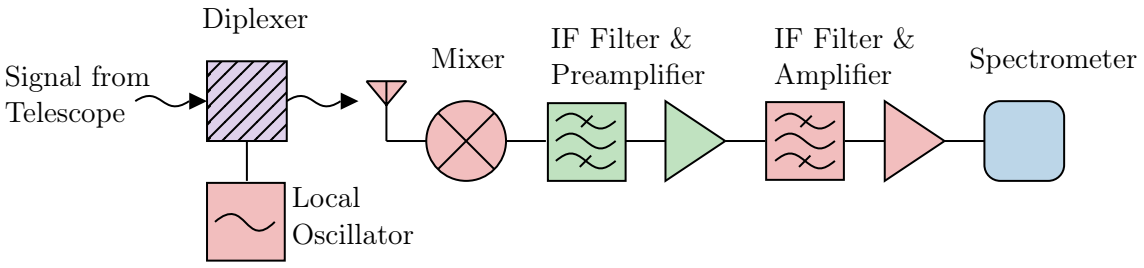


Figure 2.6.: Typical heterodyne receiver
The signal received by the telescope is combined with the LO signal in a diplexer. An antenna couples the resulting free space EM waves to an electron current which is fed into a nonlinear mixer. The mixer creates the intermediate frequencies of which the desired intermediate frequency is then filtered and amplified and finally digitized by a spectrometer.

Antenna Theory

An antenna is an electrical device to convert an electrical current into free space EM radiation and vice versa. Generally, an antenna can only receive or transmit radiation of one polarization

direction. Numerous different antenna types are available with the dipole antenna being the simplest form. Each type exhibits a different beam pattern $F(\theta, \Phi)$ that describes the directional dependence of the strength of the emitted waves from the antenna. In the case of the dipole, the beam pattern will take the shape of a donut. Figure 2.7 shows an arbitrary antenna pattern that exhibits several side lobes additional to the main beam. Tapering the main beam helps to reduce the influence of the side lobes but reduces also the antenna's resolution. The antenna forward efficiency η_{fss} is a measure for the ratio of radiation received from the forward hemisphere to the total received radiation. This efficiency is typically close to one.

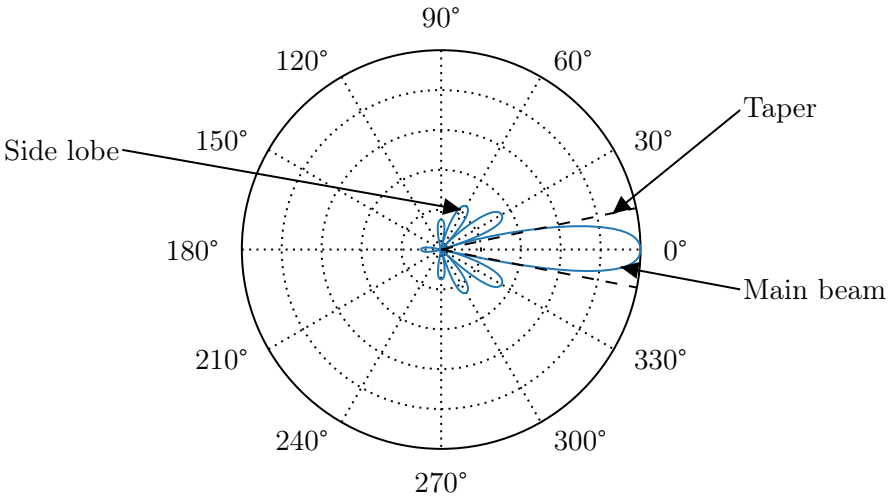


Figure 2.7.: Exemplary antenna beam pattern
Typically an antenna beam pattern exhibits a strong forward-directed main beam as well as some side lobes. Some of these side lobes might be even backward-directed. Tapering the beam like indicated by the dashed lines can significantly reduce the effect of the side lobes but it also reduces the antenna's resolution

An antenna receives a PSD P_ν from its environment that can be converted into a noise temperature using equation (2.7). This can be motivated by the *gedankenexperiment* in figure 2.8, where an antenna is completely contained in an enclosure of the temperature T_B and terminated by a matching resistor of temperature T_A. The second law of thermodynamics requires a thermal equilibrium between the antenna and the terminating resistor and therefore leads to $T_A = T_B$. This said the antenna temperature T_A is a measure of the power received by an antenna.

Mixer

The mixer is the centerpiece of the receiver and plays an important role as it is responsible for the conversion of the sum of the two input signals into the frequency-shifted signals. To make this possible the mixer must exhibit a non-linear characteristic $U_{out} = F(U_{in})$ where the output voltage U_{out} is dependent on the input voltage V_{in} in a non-linear way. This non-linear characteristic can be written in terms of a Taylor expansion[13]

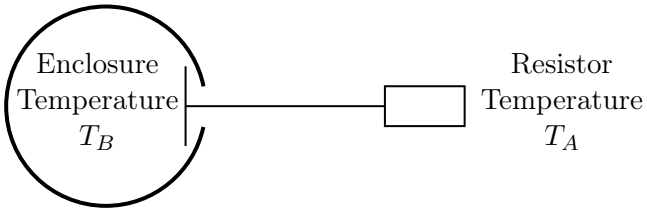


Figure 2.8.: An antenna is enclosed by a black body radiator of temperature T_B and connected to a terminating resistor of temperature T_A[6]. The second law of thermodynamics requires $T_A = T_B$.

$$\begin{aligned} F(U_{in}) &= F(0) + \frac{\mathrm{d}F}{\mathrm{d}U} \cdot U_{in} + \frac{1}{2} \cdot \frac{\mathrm{d}^2F}{\mathrm{d}U^2} \cdot U_{in}^2 + \frac{1}{6} \cdot \frac{\mathrm{d}^3F}{\mathrm{d}U^3} \cdot U_{in}^3 + ... \\ &= K_0 + K_1 \cdot U_{in} + K_2 \cdot U_{in}^2 + K_3 \cdot U_{in}^3 + ... \end{aligned} \tag{2.38}$$

Using only the terms up to the second-order and $U_{in} = A \cdot \sin(2\pi\nu_{LO}t) + B \cdot \sin(2\pi\nu_{sig}t)$ with the signal frequency ν_{sig} and the local oscillator frequency ν_{LO} yields

$$\begin{aligned} U_{out} &= K_0 + K_1 \cdot (A \cdot \sin(2\pi\nu_{LO}t) + B \cdot \sin(2\pi\nu_{sig}t)) \\ &\quad + K_2 \cdot (A \cdot \sin(2\pi\nu_{LO}t) + B \cdot \sin(2\pi\nu_{sig}t))^2 + H.O.T. \\ &= K_0 + K_1 \cdot (A \cdot \sin(2\pi\nu_{LO}t) + B \cdot \sin(2\pi\nu_{sig}t)) \\ &\quad + K_2 \cdot \left(\frac{A^2}{2} + \frac{B^2}{2}\right) - K_2 \cdot \left(\frac{A^2}{2} \cdot \cos(4\pi\nu_{LO}t) + \frac{B^2}{2} \cdot \cos(4\pi\nu_{sig}t)\right) \\ &\quad + K_2 \cdot A \cdot B \cdot (\cos(2\pi(\nu_{LO} - \nu_{sig})t) - \cos(2\pi(\nu_{LO} + \nu_{sig})t)) + H.O.T. \end{aligned} \tag{2.39}$$

The last term of equation (2.39) contains the aforementioned sum and difference of the two input signals. Strictly speaking, the difference $\nu_{LO} - \nu_{sig}$ produces positive frequencies for $\nu_{LO} > \nu_{sig}$ and non-physical negative frequencies for $\nu_{LO} < \nu_{sig}$. However, only the absolute value of the frequency can be observed in reality thus leading to an interesting phenomenon. A feature like a line emission in the IF band may originate from two bands in the incoming signal which are located symmetrically around the local oscillator frequency. The band with the lower frequency is called the lower sideband (LSB) and the other the upper sideband (USB). Sometimes LSB and USB are also referred to as signal band and image band and vice versa, depending on which of these two bands contains the region of interest. This situation is depicted in figure 2.9 which shows how the LSB and the USB are summed to form the dual sideband (DSB) signal. Such a receiver is called a DSB heterodyne receiver, whereas a single sideband (SSB) receiver filters for either the USB or the LSB before mixing. Because of the conservation of energy, the amplitudes in the IF are only half of the original amplitudes. Depending on the receiver, the IF amplitudes are sometimes calibrated to the SSB amplitude to preserve the line intensities as heterodyne detectors are mostly used for line observations. The downside of this proceeding is that the continuum temperature which is present in both the LSB and the USB is now twice its real temperature. This will be of importance for the simulation of a Herschel observation in section 6.2.

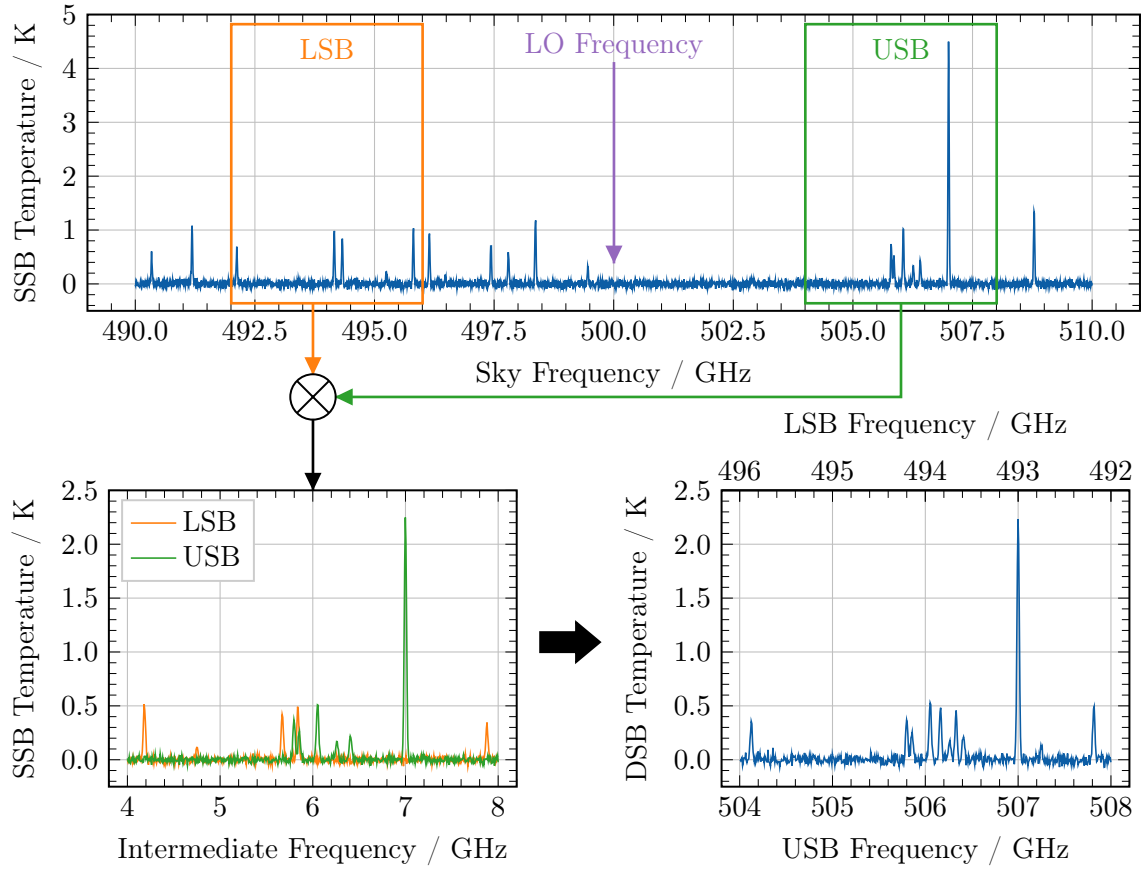


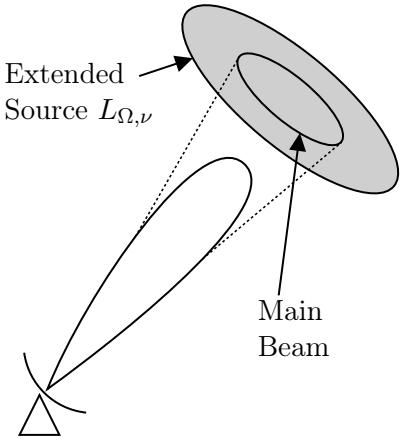
Figure 2.9.: The ambiguity of the DSB signal
The mixer shifts the signals from the LSB and the USB to the IF band creating two SSB signals. However, only the sum of both SSB signals can be measured as DSB signal.

A distinction between the LSB and the USB requires some special techniques. Shifting the LO frequency results in features from the LSB being shifted in a different frequency direction than features from the USB, because the LSB is mirrored in the IF. By carrying out the same observation with slightly different local oscillator tunings, a technique called deconvolution[14] can be used to separate the two bands.

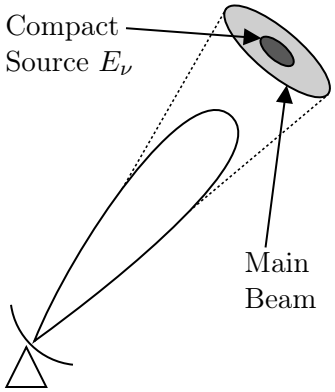
Measurement

"Because of the way a heterodyne receiver is calibrated (i.e. measuring the receiver temperature T_{rec}, with a hot and a cold load), the logical intensity unit for a heterodyne observation is temperature, expressed in Kelvin (K)"[7]. Usually, the antenna temperature T_A is used.

For the detailed calculation of the antenna temperature, the two cases of a point source and an extended source need to be distinguished as shown in figure 2.10.



(a) Extended source
The observable $L_{\Omega,\nu}$ can be measured.



(b) Compact source (e.g. point source)
Only the observable E_ν can be measured. This effect is called beam dilution.

Figure 2.10.: The two figures show the main lobe of the antenna enclosed by the dotted FOV of the telescope for the two cases of the source's size[6].

Point Source The PSD received by an antenna can be expressed as[7]

$$P_\nu = k_B T_A = \frac{1}{2} \cdot E_\nu \cdot A_{eff} \cdot \frac{1}{\eta_{fss}}. \tag{2.40}$$

with the factor $\frac{1}{2}$ taking account for one polarization per receiver and the effective antenna area A_{eff}. This can be calculated by

$$A_{eff} = \eta_{ap} \cdot A_{ap} \tag{2.41}$$

where the aperture efficiency η_{ap} takes the antenna's efficiency for measuring point sources into account. Rearranging equation (2.40) yields an expression for the calculation of the antenna temperature in the case of a point source:

$$\boxed{T_A = \frac{E_\nu \cdot A_{eff}}{2 \cdot k_B \cdot \eta_{fss}}} \tag{2.42}$$

Extended Source In the case of an extended source, the antenna is assumed to be tapered in such a way that only the main beam is exposed to the extended source. This is a reasonable assumption that is satisfied by the two prominent heterodyne detectors GREAT and HIFI. This said the main beam temperature T_{mb} equals the Rayleigh-Jeans brightness temperature of the extended source

$$T_{mb} = T_B = L_{\Omega,\nu} \cdot \frac{c^2}{2 \cdot k_B \cdot \nu^2} \tag{2.43}$$

The main beam temperature can be converted to an antenna temperature using equation 5.4 from [15]

$$T_{mb} = \frac{\eta_{fss}}{\eta_{mb}} \cdot T_A, \tag{2.44}$$

with the main beam efficiency η_{mb}. This efficiency is defined as the ratio of the main beam solid angle

$$\Omega_{mb} = \iint\limits_{mb} F(\theta, \Phi)\ \mathrm{d}\theta\, \mathrm{d}\Phi \tag{2.45}$$

to the antenna solid angle

$$\Omega_A = \iint\limits_{4\pi} F(\theta, \Phi)\ \mathrm{d}\theta\, \mathrm{d}\Phi = \Omega_{mb} + \Omega_{sl} \tag{2.46}$$

with the sidelobe solid angle Ω_{sl}, representing the fraction of power coming from the main beam compared to the overall received power[13]

$$\eta_{mb} = \frac{\Omega_{mb}}{\Omega_A}. \tag{2.47}$$

Putting these equations together leads to an expression for the computation of the antenna temperature for an extended source:

$$\boxed{T_A = L_{\Omega,\nu} \cdot \frac{c^2}{2 \cdot k_B \cdot \nu^2} \cdot \frac{\eta_{mb}}{\eta_{fss}}} \tag{2.48}$$

Equation (2.48) can also be derived in a very similar way to the point source case. The PSD received by an antenna from a uniformly illuminated extended source that fills the complete main beam can be calculated by

$$P_\nu = k_B T_A = \frac{1}{2} \cdot L_{\Omega,\nu} \cdot \Omega_{mb} \cdot A_{eff} \cdot \frac{1}{\eta_{fss}}. \tag{2.49}$$

Rearranging this equation results in an expression for the antenna temperature which requires the knowledge of the main beam solid angle Ω_{mb}

$$T_A = \frac{L_{\Omega,\nu} \cdot \frac{\Omega_{mb}}{\Omega_A} \cdot \Omega_A \cdot A_{eff}}{2 \cdot k_B \cdot \eta_{fss}} \tag{2.50}$$

The antenna theorem is given without derivation from [6] as

$$\Omega_A = \frac{\lambda^2}{A_{eff}} = \frac{c^2}{A_{eff} \cdot \nu^2} \tag{2.51}$$

which relates the angular antenna resolution to the effective antenna area and the observed wavelength. Plugging the antenna theorem (2.51) and the definition of the main beam efficiency (2.47) into equation (2.50) yields again equation (2.48).

SNR Calculation

The calculation of the SNR of an observation with a heterodyne instrument requires a value for the observation's noise. As the signal T_A is measured as Temperature in Kelvins, the noise ΔT_{rms} is also measured as noise temperature following eq. (2.7).

The system temperature T_{sys} of the detector is a measure for the total detected power including the astronomical signal T_A as well as background radiation sources T_{bg} and the disturbances of the detector's electronic components captured by T_{rec}

$$T_{sys} = T_{A,LSB} + T_{A,USB} + T_{bg,LSB} + T_{bg,USB} + 2 \cdot T_{rec}. \tag{2.52}$$

The factor of 2 takes account for the receiver temperature being the same in both the upper and lower sideband[7]. Assuming a Gaussian distribution of the noise which holds according to the central limit theorem[6], the root mean square (RMS) noise temperature ΔT_{rms} of N averaged data points can be calculated as

$$\Delta T_{rms} = \frac{T_{sys}}{\sqrt{N}}. \tag{2.53}$$

The number of data points N taken during an integration time t_{exp} is given by the Nyquist-Shannon theorem with the signal bandwidth $\Delta\nu$

$$N = \Delta\nu \cdot t_{exp}. \tag{2.54}$$

Equation (2.53) and (2.54) yield for the RMS noise temperature

$$\Delta T_{rms} = \frac{T_{sys}}{\sqrt{\Delta\nu \cdot t_{exp}}}. \tag{2.55}$$

This equation is generally known as the *radiometer equation*. However, this equation does not take the chopping between the astronomical source and the background into account which is necessary to remove the background radiation during data processing. As chopping reduces the effective exposure time by a factor of $\frac{1}{2}$ and the subtraction of two independent random variables (signal and background) increase the noise by a factor of $\sqrt{2}$, the equation for the RMS noise temperature becomes

$$\Delta T_{rms} = \frac{2 \cdot T_{sys}}{\sqrt{\Delta\nu \cdot t_{exp}}}. \tag{2.56}$$

The SNR of the observation can now be calculated as

$$\mathrm{SNR} = \frac{T_A}{\Delta T_{rms}} = \frac{T_A}{2 \cdot T_{sys}} \cdot \sqrt{\Delta\nu \cdot t_{exp}}. \tag{2.57}$$

Due to the limited spatial resolution of heterodyne detectors, an observation technique called on-the-fly mapping is used to map larger regions. Thereby, "the telescope scans along a series of rows while the back-ends are continuously integrating the incoming signal"[7]. The SNR of such an observation is given by

$$\mathrm{SNR} = \frac{T_A}{T_{sys} \cdot \sqrt{1 + \sqrt{\frac{1}{N_{on}}}}} \cdot \sqrt{\Delta\nu \cdot t_{exp}} \tag{2.58}$$

where t_{exp} "is the on source exposure time per point, and N_{on} is the number of on-source positions between measurements of the reference position"[7].

3. Preparatory Work

After all the required mathematical equations for the exposure time calculator have been collected, the design and implementation of the new software can be started. Some useful tools support the developer in this lengthy process to stay on top of things which are explained in the following. But first, a brief introduction to python is given.

3.1. Python

Python was developed in 1991 by Guido van Rossum in Amsterdam as an open-source scripting language for the operating system *Amoeba*. The goal was to develop a simple and easily readable but powerful programming language. It supports both functional and object-oriented programming as well as other programming paradigms. Even though Python is an interpreted scripting language, every program needs to be compiled to byte code which can then be interpreted by the Python interpreter. Since the not backward compatible version 3.0, Python was completely redesigned and offers a garbage collection for efficient memory management. Today, Python is managed by the *Python Software Foundation* which coordinates the open-source development. Thanks to the portability and the many package extensions, the language is among the most popular programming languages in the world[16]. For these reasons, Python was chosen for the implementation of ESBO-ETC .

3.2. Project Structure

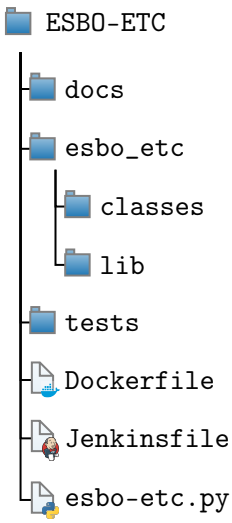


Figure 3.1.: Project Structure

Prior to the implementation of the software, the project's folder structure has to be defined. A typical python project structure was chosen for ESBO-ETC as shown in figure 3.1. The folder `docs` contains all source files of the documentation as well as all necessary scripts for building the documentation (see also section 3.4.3). The Python source code is stored in `esbo_etc` with all classes in the subfolder `classes` and all modules in `lib`. This structure enables hierarchical imports in the software's main file `esbo-etc.py`. Besides the source code of the documentation and the software, the project also makes use of build tests for the software as described in section 3.4.2. The source files of these tests are contained in `tests`.
Finally, the project's root directory also contains a `Dockerfile` which defines the build-environment of the CI/CD pipeline defined in `Jenkinsfile` (see also section 3.3.3 and 3.3.2).

3.3. Tools

Several tools and libraries were used to support and simplify the process of software development. This includes on the one hand tools for source code management and deployment like *git*, *Jenkins* or *docker* but also extensive Python packages for testing and documentation.

3.3.1. Version Control

Version control refers to a system for managing and tracking changes to computer files, mostly programming source code. Every change and the responsible person can be traced and, in case of an error, reverted. Furthermore, it allows the collaboration of many people on the same project or even the same file by merging all applied changes. The most popular version control system *git* was used for ESBO-ETC .

git

Unlike many other version management tools, *git*, which was originally developed by Linus Torvalds for his Linux kernel, initially works locally on the development computer. A repository is created for each project to capture its revision history. Files can be added to the tracking and changes including a brief description of the applied changes can be committed to the repository. In a second step, the changes in the local repository can be pushed to a server repository, where the changes of all developers can be merged. The other way round, the changes of other developers can be pulled from the server repository to the local repository. There are numerous web interfaces available for managing the server repository, of which *GitHub* is probably the best known. In this case, however, a local installation of *Gitea* was used as this is the preferred software of the IRS.

Gitea

Gitea is an open-source version management software based on git. It offers a sophisticated web GUI for administrating and browsing the server-side repository. Besides the repository management, Gitea offers several other features, such as the collection of issues, release management as well as a wiki for documentation. Another useful feature is the integration of webhooks which will be triggered after a push to the repository. This mechanism is required for CI/CD tools like *Jenkins* in section 3.3.2. The ESBO-ETC repository is publicly accessible at `https://egit.irs.uni-stuttgart.de/esbo_ds/ESBO-ETC`.

3.3.2. Jenkins

Jenkins is an open-source automation server for tasks during software development like testing, building or deploying applications. This automation flow is called a continuous integration / continuous development (CI/CD) pipeline. All steps, tasks and artifacts of the pipeline are defined in a file called `Jenkinsfile` in the JSON format. After the installation of Jenkins, a new pipeline can be set up by establishing a connection to a source code repository containing a `Jenkinsfile`. By adding a webhook to the repository, the pipeline is triggered after each push to the repository. During each run of the pipeline, all stages are executed on the Jenkins-server using the environment defined in the `Jenkinsfile`.

3.3.3. docker

docker is an open-source software that allows individual applications to run in isolated and portable docker containers. docker is deeply integrated into the Linux kernel, which is why docker containers can be executed directly at the kernel level as processes.
Before a docker container can be executed, a so-called docker image must be built, which is a blueprint of a docker container. A docker container can be derived from this image on any system, independent of the environment. Such a docker image is built using a `Dockerfile`, which defines the steps for building the docker image. Each of these steps is executed sequentially in a separate docker container based on the temporary docker image of the previous step. The result of each step is added as a new layer to the target image.
Although it is possible to create a docker image from scratch, in most cases the extension of an existing docker image is much more efficient. For this purpose, each `Dockerfile` contains the instruction `FROM` at the beginning, defining the parent docker image. An example of this is the docker image *Alpine*, which contains a minimal installation of the Linux distribution Alpine and thus already contains numerous useful applications such as a shell as well as some libraries.

3.4. Python Packages

One reason Python was chosen for the implementation of ESBO-ETC is the large amount of freely available packages that can be used. Among many others there exist packages for astronomic unit-based calculations like *Astropy*, packages for easy documentation writing like *Sphinx* as well as for testing source code like *unittest*. Even though some more *de facto* standard packages like *numpy* or *scipy* have been used for ESBO-ETC , only the rather unusual packages are described in the following.

3.4.1. Astropy

The *Astropy* project is a community-driven collection of libraries for astronomical calculations in Python. It contains numerous methods for data structure and transformation, file input and output as well as computations. Besides the provided universal constants like h, c or k_B, primarily the tools for reading FITS and ASCII files, as well as the units-module have been used for ESBO-ETC .
Especially the units-module proved to be very helpful as it allows efficient unit-based calculations in python. This prevents bugs due to wrong unit prefixes (like milli, micro or nano) or incorrectly implemented formulas. In detail, Astropy not only checks the units for consistency during calculations but also allows the conversion of units as well as simplifications. It also allows defining the units of a method parameter in order to ensure correct method arguments. These units are also used for the XML configuration file as well as the column headers of input files.

3.4.2. unittest

In software development, unit testing describes the process of testing individual units of an application to verify the correctness of the source code. To prevent breaking already working modules, the Python package *unittest* was used to execute automated build tests.
unittest allows to define and execute test cases including both start-up scripts and shut-down scripts. After a test has been executed, the result is compared to a default value. Throughout this thesis, a total of 47 unit tests have been defined which test all classes of ESBO-ETC against stored results. As explained later in section 4.6, these tests are part of the build-pipeline's first stage. In addition, these tests may be executed manually during the development at any time to check for any breaking changes.

3.4.3. Sphinx

The Python framework *Sphinx* was used for building the documentation of ESBO-ETC . It provides the necessary tools to compile documentations in HTML or LaTeX from source code using the *reStructuredText*-syntax. Using reStructuredText enables powerful markup commands like cross-references, embedded images, text styling and many more. As the documentation is built from these source files, they can be version controlled as well using the methods described above.
Besides the manual writing of the documentation, the plugin *napoleon* was used to create an API documentation from the docstrings of all developed Python classes and methods.
The documentation of ESBO-ETC can be either manually compiled or automatically by the project's CI/CD-pipeline as explained in section 4.6. This pipeline not only compiles the documentation to HTML but also deploys the latest version to the webroot of ESBO's webserver.

4. Implementation

All properties and features of the developed software are explained in this chapter starting with some details on ESBO-ETC's architecture. This is followed by an explanation of all numerical approaches to mathematical equations. Finally, the input and output files as well as the documentation are presented.

4.1. Software Architecture

Choosing the right architecture is crucial for the implementation and expandability of the software. As python natively supports object-oriented programming (OOP), many popular design patterns for OOP can be used. These patterns help to fulfill the two general rules for OOP:

1. **Open-closed principle:** "software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification"[17]. This means, that software should be extendable without having to modify the existing source code.

2. **Cohesion vs. coupling:** The term *coherence* refers to how specialized a class is designed. A class with many different tasks has low cohesion. The interdependency of classes is described by the term *coupling*. Classes have a high coupling if changes to one class directly require modifications to the other classes. Ideally, software has a strong cohesion and a low coupling of the classes.

To satisfy both rules, the *decorator* pattern and the *factory* pattern have been chosen for ESBO-ETC and will be explained in the following sections. The parameters of all methods are omitted in the following for the sake of clarity.

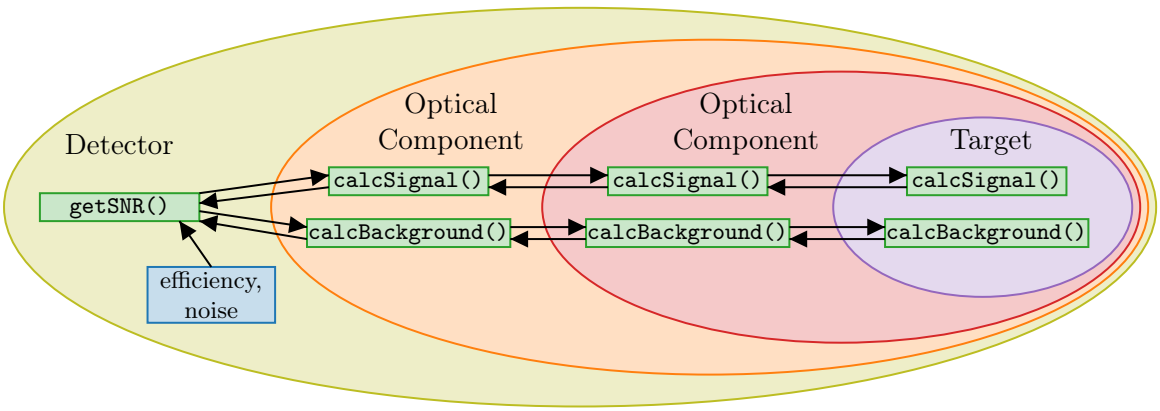


Figure 4.1.: Decorator pattern used for ESBO-ETC
The methods `calcSignal()` and `calcBackground()` are called in a cascade from the outermost object to the innermost object.

4.1.1. Decorator Pattern

ESBO-ETC aims at simulating the emission, transfer and detection of electromagnetic radiation. All involved components in this unidirectional process are defined dynamically in the XML configuration file. This situation can be modeled perfectly using the decorator pattern. This pattern allows to dynamically attach "additional responsibilities to an object"[18] by decorating the object with another object providing the additional features. The decorating object, therefore, has to have the same methods as the decorated object and forwards any method call to the decorated object before returning the result. The use case of the decorator pattern for ESBO-ETC is shown in figure 4.1.
An astronomical target is always the start of the radiative transfer process and is therefore always the core object of the decorators. It might be decorated by any optical component like an atmospheric model or a mirror component that requests the signal and background radiation from the decorated object. Before returning the quantity, an optical component may increase or decrease the radiation by extinction or emission. Just like in reality a detector forms the end of the radiative transfer and can therefore not be decorated.

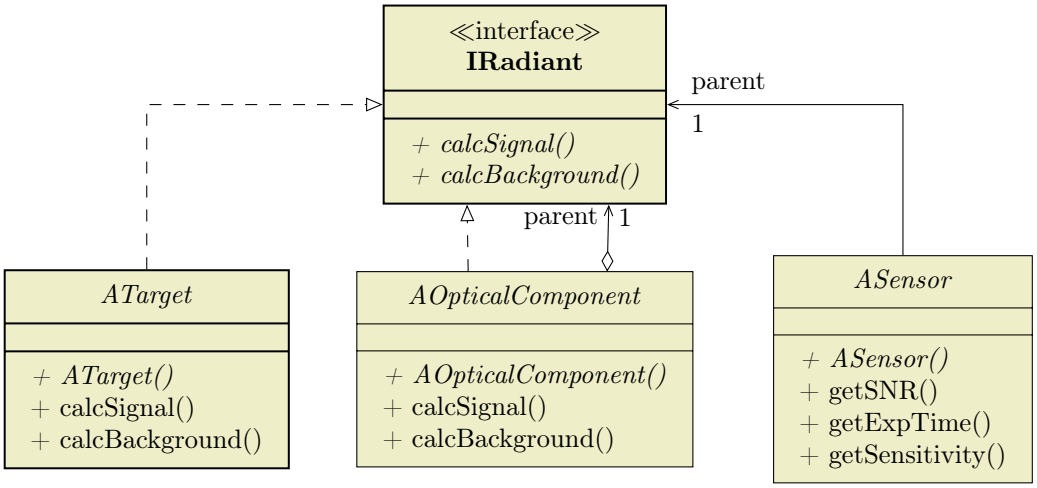


Figure 4.2.: UML representation of the decorator pattern used for ESBO-ETC
The interface `IRadiant` is implemented (dashed arrow) by every decorable class (`ATarget`, `AOpticalComponent`). Any decorator decorates an object of the type `IRadiant`. Finally, an `ASensor`-object possesses (solid arrow) an object of type `IRadiant` but is not decorated by any other class. All shown classes are abstract because they are prototypes for the actual components.

This architecture can be represented as Unified Modeling Language (UML) class diagram as shown in figure 4.2. The interface `IRadiant` defines the methods to be implemented by all decorable classes. As explained in figure 4.1, these methods are `calcSignal()` and `calcBackground()`. All target classes shall be able to be decorated and therefore implement the interface `IRadiant`. Furthermore, all optical components shall be able to decorate any class with the interface `IRadiant` but also be decorated. This is the reason why the abstract superclass for all optical components `AOpticalComponent` implements the interface and possesses an object with the interface `IRadiant`. This situation is represented by the implementation arrow and the unidirectional association of the class `AOpticalComponent` in

figure 4.2. Eventually, a sensor-object terminates the series of decorated objects which is the reason why the abstract class `ASensor` only possesses an object with the interface `IRadiant` but doesn't implement the interface itself.
The selected architecture for the radiative transfer allows to dynamically add optical components of different kinds between the astronomical target and the detector. Each component may alter the incident radiation in its own way independent from the previous components. Furthermore, future targets, optical components and detectors can be added easily, as they are independent of the other classes and just have to follow the decorator pattern. Thereby, both rules of OOP are satisfied.

4.1.2. Factory Pattern

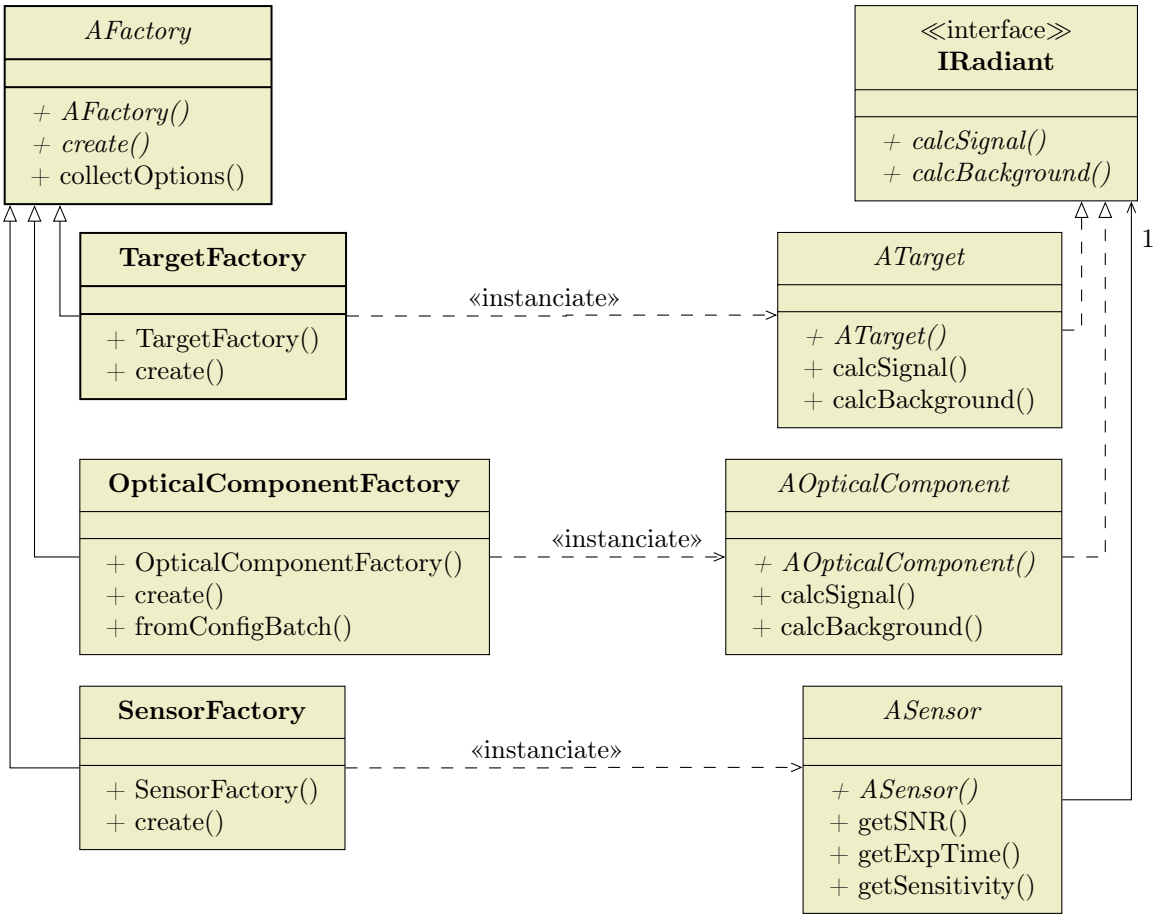


Figure 4.3.: UML representation of the factory pattern used for ESBO-ETC
The abstract class `AFactory` defines the abstract method `create` for the creation of new elements which is implemented by all subclasses. These subclasses are responsible for the instantiation of the subclasses of the interface `IRadiant`. Even though `ASensor` is not implementing the interface `IRadiant` and therefore doesn't strictly follow the factory pattern, its factory class `SensorFactory` is still a subclass of `AFactory` and therefore included in the class diagram.

The factory pattern is used in OOP to define "an interface for creating an object, but let subclasses decide which class to instantiate"[18]. This said, the factory pattern allows to add new components without having to modify the existing factories and fulfills thereby the open-closed principle of object-oriented programming. Besides that, the factory pattern increases to code's coherence, as the instantiation and usage of objects are strictly separated.
Figure 4.3 shows the factory pattern as it is used for ESBO-ETC . The abstract superclass `AFactory` defines the method `create()` which has to be implemented by all subclasses for the construction of the corresponding component. Each sort of component of the radiative transfer and detection process (target, optical component and detector), are created by a separate factory. As the class `ASensor` does not implement the interface `IRadiant`, it doesn't strictly follow the factory pattern. Yet it is shown in the diagram as its corresponding factory `SensorFactory` inherits from the superclass `AFactory`.
The factory template `AFactory` additionally provides the method `collectOptions()` to all factories which allows collecting the required instantiation parameters of the components from the configuration file.
It has to be mentioned, that not the abstract classes `ATarget`, `AOpticalComponent` and `ASensor` are instantiated by the corresponding factories but rather their subclasses. These subclasses have been omitted in the class diagram for the sake of clarity.
The method `fromConfigBatch` of the factory `OpticalComponentFactory` allows creating multiple optical components at once from a batch of configuration entries.
To reduce the coupling of the source code, the three factories shown in figure 4.3 do not contain specific code for any class that can be instantiated. This proceeding enforces the open-closed principle as well, due to the fact that new subclasses of the abstract classes on the right-hand side of figure 4.3 can be added without having to modify the corresponding factory.

4.1.3. Imaging Detector

The imaging detector is implemented as subclass of `ASensor` as shown in the class diagram in apendix D. The architecture shown in figure 4.4 has been developed to facilitate the computation of the SNR for the imaging detector. First of all, the class `PixelMask` derived from the numpy `ndarray` enables a representation of the detector's 2D pixel array. Besides the inherited methods, this class takes the pixel properties like the size or the center position into account and allows placing a virtual photometric aperture on the grid.
The interface `IPSF` defines methods for working with point spread functions: The method `calcReducedObservationAngle` allows calculating the radius of a virtual photometric aperture as observation angle reduced by $\frac{D_{ap}}{\lambda}$. This reduced observation angle $\bar{\theta}$ can be used for the realization of the photometric aperture on the pixel mask as explained above. Besides the calculation of the photmetric aperture's diameter, the interface `IPSF` defines the method `mapToPixelMask()`. This method is intended for integrating the PSF on the pixel grid. Thereby, the incident radiation's distribution on the pixel grid is determined according to equation (2.34). Both methods intrinsically take jitter into account for the calculation.
The first version of ESBO-ETC as delivered with this thesis includes a class for handling PSFs described by the airy disk (see section 2.3). This shape is the default and will be used if no additional keys are given in the configuration file. In contrast to the other two PSF representation, the alteration of the PSF due to FOV obstruction is taken into account. The

abstract class `AGriddedPSF` provides some methods for working with Point spread functions that are given as 2D grid. On the one hand, the grid can be read from an output file of the software *Zemax* by the class `Zemax` or it can be read from a FITS-image by the class `FITS`. However, both classes cannot take the FOV's obstruction into account. Nevertheless, both classes include the effect of pointing jitter.

4.1.4. Class Diagram

The full class diagram as shown in appendix D consists mostly of the classes mentioned before and their corresponding subclasses. Besides these classes, the class `SpectralQty` allows calculations and mathematical operations with spectral quantities. The capabilities of this class are explained later in section 4.2.1.

4.2. Numerical Approaches

Due to the finite precision of computers, numerical methods must be used to some extent when implementing mathematical equations. For example, computers cannot handle continuous spectral quantities as used in chapter 2 which is the reason why the class `SpectralQty` was developed to handle these quantities. On the other hand, some mathematical equations like the integration of the PSF cannot be solved analytically and require therefore numerical approaches. This applies especially to the distribution of the incident radiation onto the imager's pixel grid according to the PSF and the pointing jitter. These approaches are explained in the following.

4.2.1. Spectral Quantity

The class `SpectralQty` was developed to handle spectral quantities like the spectral radiance of an astronomical target. The spectral quantity is stored discretized in two arrays: one holding the wavelength bins and one holding the corresponding values of the spectral quantity. The resolution of the grid is controlled by the configuration file tags `wl_delta` or `res` respectively. To allow the usage of these spectral quantities in mathematical equations, the most common mathematical operations are implemented using python's magic methods. These magic methods are identified by two leading and two trailing underscores e.g. `__add__()` and are invoked implicitly by Python. The magic method `__add__()` for example is invoked for the left-hand side object in an addition. If the operation is not supported by the left-hand side object, the method `__radd__()` of the right-hand side object is called. The addition, subtraction, multiplication, division, equality and power operators are implemented using these magic methods. Depending on the operation, the other object used for the operation might be a number, a unit-based quantity or another spectral quantity.
Before a mathematical calculation with two spectral quantities can be carried out, their wavelength binning has to match. Therefore, the method `rebin()` allows changing the binning of a spectral quantity using linear interpolation. By default, extrapolation is disabled and zero is returned for wavelengths outside of the original wavelength range. This setting can be overridden to allow extrapolation of the requested values or to truncate the requested values to the valid range.
The calculation of the collected electrons of a CCD detector requires the integration of a spectral quantity as shown in equation (2.17. This operation is realized in the method

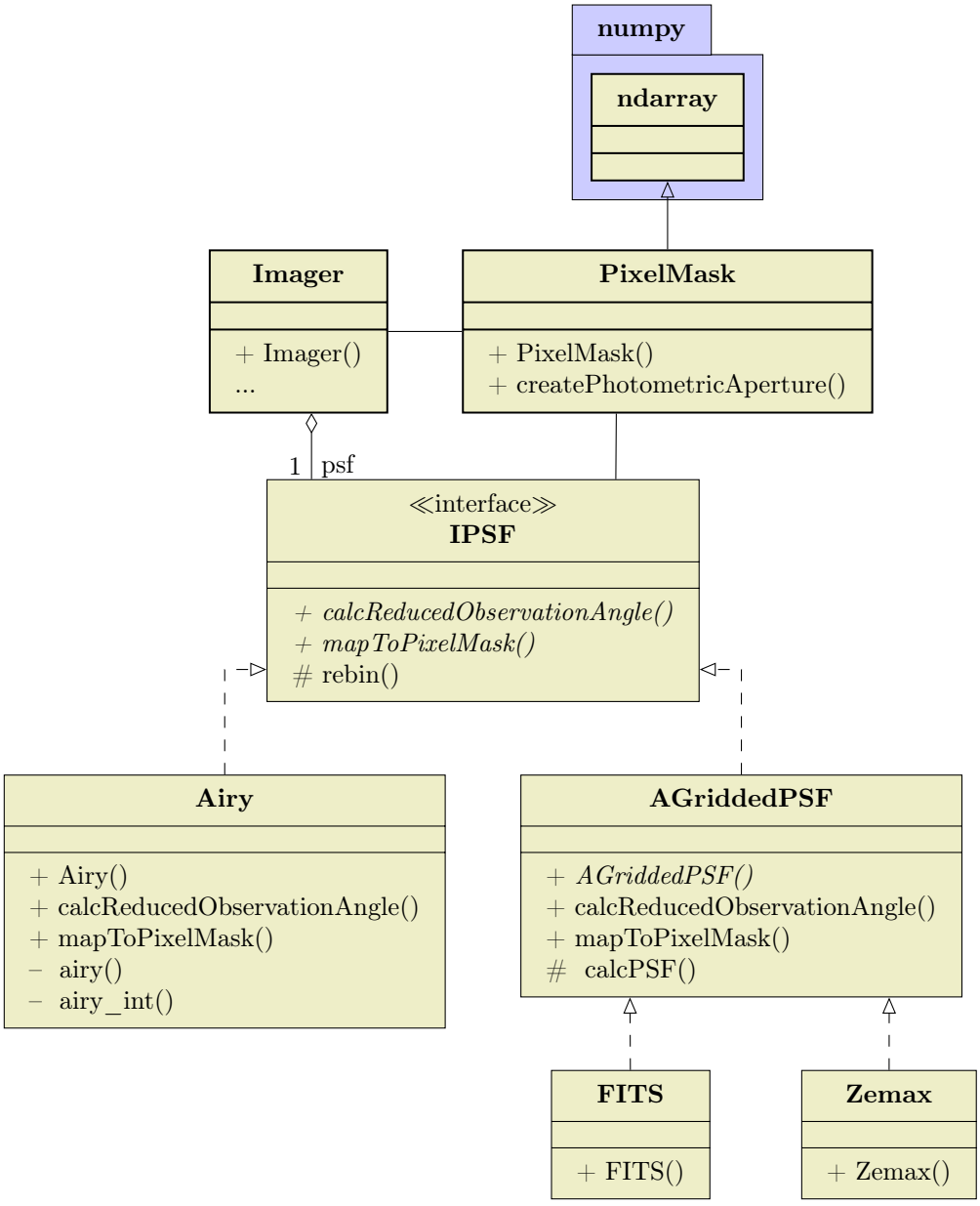


Figure 4.4.: UML representation of the architecture used for the distribution of the incident radiation on the detector pixels.
Derived from the *numpy*-class `ndarray`, `PixelMask` is used to model the detector's pixel grid. Furthermore, it allows setting up a virtual photometric aperture on the array. The interface `IPSF` defines methods for working with PSFs like the calculation of the photometric aperture's diameter as well as mapping a PSF onto the pixels.

`integrate()`) which uses the trapezoidal rule (the simplest Newton–Cotes formula) for the integration. The spectral quantities used in ESBO-ETC depend on the input files and are therefore generally unknown. The trapezoidal rule as an integration rule of first-order inte-

grates even discontinuous function with a low error. For this reason and the low computational effort, this rule was chosen.
As a result of the discretization of spectral quantities, the accuracy of the calculations for the imaging detector depends on the defined spectral resolution in the configuration file. It is therefore up to the user to balance between computational effort and precision.

4.2.2. Imaging Detector

The calculation of the incident radiation's distribution onto the pixels of the imaging detector requires the use of numerical methods. First of all, the diameter of the photometric aperture has to be computed by solving equation (2.35) for the diameter of the photometric aperture D_{phot}. Afterward, the photometric aperture has to be mapped onto the pixel grid requiring a rasterizing algorithm. The amount of incident radiation per pixel can be calculated in the end by integrating the PSF on a per-pixel basis. The approaches used to solve these problems are described in this section. Common to all approaches is the usage of a pixel oversampling factor n_{osf} defined in the configuration file to control the resolution of all numerical calculations. By default, $n_{osf} = 10$. Higher oversampling factors may enhance the precision of the calculations but also increase the computational effort by $\mathcal{O}\left(n^2\right)$ in some cases.

Calculation of the Photometric Aperture Size

As explained in section 2.4.1, the SNR of the imaging detector is not calculated on a per-pixel basis but rather for a defined virtual photometric aperture. If not defined in the configuration file, the size of this aperture has to be calculated (numerically) using the given encircled energy or keyword. The effects of pointing jitter are taken into account during the computations. Depending on the used PSF, different computation methods are used as explained in the following.

Airy Disk The airy disk representation of the PSF supports all variants of encircled energy keywords. However, pointing jitter can only be considered for a given percentage of encircled energy or the keyword *fwhm*. The calculation of the reduced observation angle consists of two steps: First of all, the reduced observation angle is calculated using equations (2.14) and (2.15). In a second step, the pointing jitter is applied and the observation angle is corrected. Due to the rotational symmetry of the airy disk, all calculations can be reduced to one dimension to reduce the computational effort.
The keyword *peak* corresponds to a virtual photometric aperture containing only a single pixel, located a the central peak of the PSF. Therefore a reduced observation angle $\bar{\theta}_{phot} = 0$ is used.
If the keyword *fwhm* is given, the full width half maximum of the airy disk is calculated. In the case of an unobstructed airy disk, the value $\bar{\theta}_{phot} = 1.028$ from the literature is used. Otherwise, the position of the FWHM is computed by solving

$$I_\lambda\left(\frac{\bar{\theta}_{\mathrm{FWHM}}}{2}\right) = \frac{I_\lambda\left(0\right)}{2} \tag{4.1}$$

for $\bar{\theta}_{\mathrm{FWHM}}$. Since the obstructed airy function is not analytically invertible, numerical methods have to be used. Since the function of the obstructed airy disk is continuous and its

analytic derivative can be calculated, Newton's method can be used for the computation of the reduced observation angle. However, this method only converges if no point with zero gradient is hit. This can be avoided by using an appropriate start value. The FWHM of the airy disk is contained between the central peak and the first minimum of which both exhibit zero gradient. For increasing obstruction ratios o, the FWHM tends towards smaller observation angles as shown in figure 2.2. Using the literature value for the observation angle of the unobstructed airy disk's FWHM, no maxima or minima with zero gradient is located between the start value and the root.

A similar situation arises with the keyword *min*: The first minimum of the unobstructed airy disk is known from the literature as $\bar{\theta}_{phot} = 2.44$. However, the position of the obstructed airy disk's first minimum has to be calculated numerically due to the irreversibility of the first derivative. Therefore the Nelder–Mead method is used to find the function's first minimum. This method is well known for its robustness but depends on a good starting point for a quick convergence. As shown in figure 2.2, the first minimum is always located between the disk's central peak and its first maximum and shifts to smaller observation angles for higher obstruction ratios. For this reason, the position of the unobstructed airy disk's first minimum is used as a starting point for the computation.

The last case covers the specification of an encircled energy percentage in the configuration file. In this case, equation (2.35) has to be solved for D_{phot} which can be converted to a reduced observation angle using the trigonometric relation $\tan\left(\frac{\theta}{2}\right) = \frac{D_{phot}}{2 \cdot f}$ by

$$\bar{\theta}_{phot} = \theta \cdot \frac{D_{ap}}{\lambda} = \frac{D_{phot}}{N \cdot \lambda}. \tag{4.2}$$

The integral of the airy disk provides an analytic solution only for the unobstructed case, all other cases have to be calculated numerically. This is done using a 21-point Gauss–Kronrod quadrature from the FORTRAN library *QUADPACK*. Additionally to the numerical integration, another numeric method is necessary to solve the equation for D_{phot}. To reduce the number of calls to the integration method, the bisection method was used. This method offers a fast convergence without having to compute the function's gradient numerically. Reducing the number of calls to the integration method to a minimum significantly reduces the computational effort as the overall calculation consists of two chained numerical computations.

After the computation of the virtual photometric aperture's diameter without taking pointing jitter into account, the diameter is corrected for the pointing jitter effect. To model this effect, the convolution of the airy disk and a Gaussian bell is calculated. Due to the radial symmetry of both the airy disk and the Gaussian bell curve, the convolution can be reduced to a one-dimensional problem to reduce the computational effort.

Again this convolution does not provide an analytic solution which is the reason why numerical convolution in the Fourier space is used. This requires both the airy disk and the Gaussian bell curve to be discretized on a grid. To cover all significant information, the grid is chosen to be of the size $2 \cdot (\theta_{phot} + 3\sigma_{jit})$ with a resolution defined by the pixel oversampling factor n_{osf} in the configuration file. By default, a resolution ten times finer than the pixel size θ_{pix} is used ($n_{osf} = 10$). With both functions mapped onto a separate grid, the convolution can be calculated. The position of the FWHM for the blurred airy disk can be found by solving equation (4.1) for $\bar{\theta}_{\mathrm{FWHM}}$ using the convolution result. On the other hand, the photometric aperture's diameter for a given percentage of encircled energy is computable

through equation (2.35) where the integration is replaced by a sum due to the discretization:

$$EE = \frac{\sum_{i=0}^{n} \left(I\left(\theta\right) \times f\left(\theta, \sigma_{jit}\right)\right)\left(\theta_i\right)}{\sum_{i=0}^{n_{osf} \cdot \left(\theta_{phot} + 3\sigma_{jit}\right)} \left(I\left(\theta\right) \times f\left(\theta, \sigma_{jit}\right)\right)\left(\theta_i\right)} \tag{4.3}$$

with $n \leq n_{osf} \cdot (\theta_{phot} + 3\sigma_{jit})$ and the resulting reduced observation angle $\bar{\theta}_{phot} = \frac{n}{n_{osf}} \cdot \theta_{pix} \cdot \frac{D_{ap}}{\lambda}$. A cumulative sum allows solving this problem efficiently.

Gridded PSF The two other available PSF representations of the ESBO-ETC version delivered with this thesis is a precomputed PSF by the software *Zemax* and a FITS-file. Zemax is a software that allows to design optical systems and calculate the system's PSF using Fourier optics which can be saved as a two-dimensional matrix of the size 32×32 in a delimiter separated file. Both representations share some methods provided by `AGriddedPSF` as shown in figure 4.4.
The gridded PSF implementation of ESBO-ETC supports only the calculation of the virtual photometric aperture's diameter from a given percentage of encircled energy using the delimiter separated file. This arises by the ambiguity of the FWHM or the first minimum of a radial asymmetric function. Unlike the calculation method used for the airy disk, the gridded PSF implementation takes the effect of pointing jitter directly into account without correcting for it in a second computational step. Therefore the PSF is parsed from the file to a two-dimensional matrix representation. According to the defined pixel oversampling in the configuration file, an oversampled PSF is calculated using two-dimensional cubic interpolation. Subsequently, this oversampled matrix is then convoluted with a Gaussian bell curve for the given jitter σ_{jit} in the frequency domain. As the result of this convolution is required for mapping the PSF onto the pixel grid in a later step (see section 4.2.2), it is stored during runtime to reduce the computation time.
After oversampling the PSF and applying possible pointing jitter effects, the photometric aperture's diameter can now be calculated from equation (2.35). Following the approach for the airy disc, the bisection method is used to find an optimal value for D_{phot}. Since equation (2.35)contains an integration over the circular virtual photometric aperture, a procedure to calculate this integral on a cartesian grid is required. In a first effort, a two-dimensional interpolation function for the function values on the grid has been defined which in turn could be plugged into a numerical integration method. However, this means that the resulting diameter of the photometric aperture is calculated by a cascade of three numerical methods: bisection, integration and interpolation. This proceeding not only gives rise to large errors but also requires a lot of computation time. A solution for this problem is the use of a rasterization of the circular aperture on the cartesian grid as described in section 4.2.2. Summing all the gridded values of the PSF contained in the rasterized circle equals the integration over the circular aperture. This way, only two numerical methods are being used: the bisection method and the circle rasterization. The precision of this computation is controlled by the user via the pixel oversampling factor n_{osf} in the configuration file.

Rasterizing the Photometric Aperture

After the diameter of the virtual photometric aperture has been calculated or defined in the configuration file, the corresponding shape has to be rasterized onto the pixel grid. This is

done by the method `createPhotometricAperture` of the class `PixelMask`. Due to the possible non-integer displacement of the photometric aperture's center from the detector center, this rasterization is an ambiguous numeric process.

Circular Aperture A conservative algorithm which generally slightly overestimates the photometric aperture was chosen for rasterizing a circular aperture onto the pixel grid. It is based on the method of Horn and illustrated in figure 4.5.

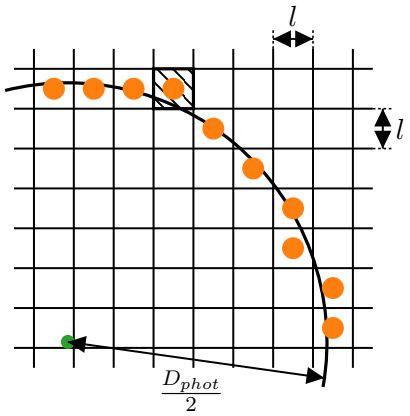


Figure 4.5.: Circle rasterizing algorithm
Each center point of the hatched pixel's border to its neighboring pixel is checked for being inside the circle. If this is true for at least one center point (which is for the center point of the southern border), the hatched pixel is marked as included in the circle.

The algorithm checks each pixel within a square of the size $D_{phot} + 2l$ for being contained in the photometric aperture. If at least one center point of the pixels borders satisfies

$$(x_{bc} - x_c)^2 + (y_{bc} - y_c)^2 \leq \left(\frac{D_{phot}}{2}\right) \tag{4.4}$$

with the circle center coordinates x_c & y_c and the border center coordinates x_{bc} & y_{bc}, the pixel is selected as a circle point. As the algorithm is repeatedly used for the computation of the photometric aperture's diameter for a gridded PSF, an efficient implementation of the algorithm is required. Since each pixel is checked independently from all other pixels, a matrix calculation using two mesh grids can be used for a fast implementation.

Quadratic Aperture The proceeding for a quadratic aperture is similar to the circular aperture. However, the computation can be reduced to a check of a single pixel in positive and negative x- & y-direction, because the edges of the square are parallel to the pixel borders.

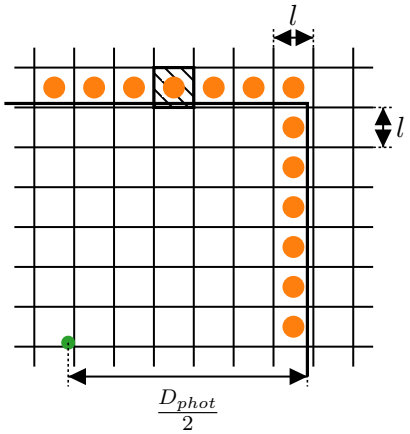


Figure 4.6.: Square rasterizing algorithm
Each center point of the hatched pixel's border to its neighboring pixel is checked for being inside the square. If this is true for at least one center point, the hatched pixel is marked as included in the square.

Integrating the PSF on the Pixel Grid

The distribution of the incident radiation on the detector pixels is determined by the telescope's point spread function and its pointing jitter. Equation (2.34) describes the calculation of the percentage of incident radiation per pixel as the integral of the convoluted PSF with the jitter function per pixel divided by the unbounded integral.
The interface `IPSF` provides the abstract method `mapToPixelMask()` for the implementation of this computation for each PSF representation. Both representations share a common proceeding for this computation: In a first effort, an interpolation function was used as an integrand for the pixel integral. However, this proved to be inefficient due to the many calls to the interpolation function during the integration. To solve this problem, the PSF is first mapped on a grid n_{osf} finer than the pixel grid. Afterward, the gridded values are being multiplied with the grid size and summed on a per-pixel basis, resulting in a numerical approximation of the PSF integral according to the Riemann sum.

4.3. Configuration File

ESBO-ETC is controlled by an XML configuration file that is parsed during the program startup. The path to this file can be passed to ESBO-ETC via a command-line parameter. It consists of the four main containers `common`, `astroscene`, `common_optics` and `instrument`. Each container may contain several tags with different parameters. A detailed explanation of all required and optional tags is given in the delivered software documentation.
Generally, each tag consists of a tag name and some parameters as keyword-value-pairs. According to the XML-standard all values have to be quoted in the configuration file regardless of the actual data type. The values are converted to the correct data type during the startup of ESBO-ETC . To avoid errors due to unit mismatches, ESBO-ETC makes use of unit-based calculations. The units of physical quantities in the configuration file can be either included in the value-string or can be defined in a second parameter with the same keyword and the suffix

_ *unit*. This situation is shown in the configuration files for the science cases in appendix F. Before running any computation, ESBO-ETC parses the configuration file and converts every XML-tag to an object of the class `Entry`. An UML representation of this class is shown in figure 4.7.
As an advantage of this conversion, each `Entry`-object now provides several check-methods to check each configuration parameter. This is done automatically by ESBO-ETC , after the configuration file has been parsed. As each tag corresponds to a python class, each of these classes independently execute the checks of their parameters. This again allows an easy extension of ESBO-ETC because the configuration check is decentralized and part of the responsibility of each class.

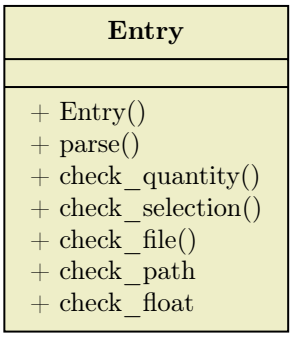


Figure 4.7.: Class `Entry` for handling XML-tags
Every XML-tag in the configuration file is parsed to an object of class `Entry`. This class provides several methods to check the validity of each parameter.

4.4. Output

The results of ESBO-ETC 's computation are written to stdout of the command-line and to output files in the output directory defined in the configuration file. In case of an error or a faulty configuration file, ESBO-ETC offers an advanced logging output to stderr.

4.4.1. Command-Line Output

After ESBO-ETC ran all computations, the results are printed to stdout of the command line. An exemplary output is shown in table 4.8. Depending on the computed quantities (SNR, exposure time or sensitivity), the layout may differ. As ESBO-ETC supports batch-computations for a list of exposure times or SNRs, the output is enumerated corresponding to the list of input values.

4.4.2. Output Files

ESBO-ETC writes the results of all computations to one or more files in the output directory defined in the configuration file. The format of the output files is controlled by the configuration file. The available formats and the content of the files depend on the used detector type as the output is written by this component. New detector components can introduce additional output formats.

```
$ ./run_esbo-etc
    ___________ ____  ____         ___________________
   / ____/ ___// __ )/ __ \       / ____/_  __/ ____/
  / __/  \__ \/ __  / / / /_____/ __/   / / / /
 / /___ ___/ / /_/ / /_/ /_____/ /___  / / / /___
/_____//____/_____/\____/     /_____/ /_/  \____/

|-----|------------------|--------------|
|  #  |   Exposure Time  |      SNR     |
|-----|------------------|--------------|
|  1  |     2.3000e+03 s |   2.7273e+00 |
|-----|------------------|--------------|
```

Figure 4.8.: Exemplary command line output for the computation of a SNR

Imaging Detector

The signal, background, read noise and dark current as total collected electrons are written as matrices to separate files. These files can be either CSV-files or FITS-files depending on the settings in the configuration file. The data written to these files is reduced to the relevant region containing the photometric aperture. Nevertheless, the reduction strategy is written to the file header to allow a lossless restoration of the original pixel matrix.

Heterodyne Detector

In the case of the heterodyne instrument, the spectral signal temperature, background temperature, RMS noise temperature in Kelvins and the SNR are written to a table in a CSV file in the output directory.

4.4.3. Logging

The verbosity of ESBO-ETC 's command line output can be controlled by supplying a log level to the command line parameter `-l / -logging`. Available logging levels are *DEBUG*, *INFO*, *WARNING* (the default) and *ERROR*. Internally, ESBO-ETC makes use of the python package *logging* for controlling the software's verbosity.
On the one hand, this approach allows to focus on the computation results and show only major program errors. On the other hand, detailed logging output like the *DEBUG* output enables the user to keep track of all computational steps and therefore to locate possible points of failure.

4.5. Documentation

An extensive documentation of ESBO-ETC is available on the ESBO website at `https://esbo-ds.irs.uni-stuttgart.de/esboetcdocs/index.html`. This documentation built with sphinx provides useful information on the installation and operation of ESBO-ETC . This documentation is intended to be the main reference for all users of ESBO-ETC . It describes the installation process on all common operating systems and explains how the software

can be run. To enable the user to build a suitable configuration file for his science case, the documentation describes all available components and their corresponding parameters in detail. Numerous configuration code snippets support this description and serve as samples for the user.
Besides the information for the end-user, the documentation contains an additional chapter for future developers of ESBO-ETC . This chapter describes ESBO-ETC 's software architecture and the used classes in detail. The extension possibilities and all required tasks are described in a step-by-step guide to enable the future development of the software. As ESBO-ETC can be used as a python module in other programs, an exemplary code snippet shows the proceeding for this.
Lastly, the docstrings of all classes and methods used in ESBO-ETC are collected in an API documentation.

4.6. CI/CD Pipeline

A CI/CD pipeline is used to ensure the proper functioning of ESBO-ETC and to simplify the deployment of new versions of the documentation. This pipeline is triggered after a commit to the project's Gitea repository and executed by a local Jenkins instance. Before any step of the pipeline is executed, a docker image containing a python installation and all necessary packages for ESBO-ETC is built. This allows running all steps of the pipeline that require a python installation independent from the host's environment. After the docker image is set up, the following three stages of the pipeline are executed:

1. First of all, the defined build tests are run in the previously created docker container.
2. If all tests passed, the documentation is being built using sphinx in the docker container.
3. Lastly, the HTML files of the documentation are copied to the webroot of ESBO's webserver.

5. Software Verification & Validation

The V-model for software development foresees a verification and validation of the newly developed software after the implementation. Even though each unit of the software is continuously tested by the unit tests in the CI/CD-pipeline (see section 4.6), a verification and validation of the software has to be conducted. During the verification, the software is checked against all previously defined requirements. Therefore, a verification scenario is defined and conducted for each software requirement. The validation on the other hand ensures that the software is well suited for its intended use.
This chapter describes the verification process and all used verification methods. All verification cases and their results are listed in appendix E. A detailed documentation of the verification process, all results and the corresponding manual calculations are given in [19]. The validation of the software is conducted as part of a science case analysis in chapter 6.

5.1. Verification Methods

ESBO-ETC is a modular exposure time calculator offering numerous different components that can be assembled like the pieces of a puzzle. It allows both the simulation of spatial as well as spectral detectors. The development was driven by the fact, that no available exposure time calculator offers these possibilities. This said, the verification of all requirements requires the use of several existing and reviewed exposure time calculators. In some cases, the verification has to rely on manual calculations as no other ETC comparably implements these requirements. This is for example the case for the consideration of the telescope's pointing jitter.
This section provides an overview of the used references for the verification and their limitations.

5.1.1. AETC Verification

The Advanced Exposure Time Calculator (AETC) is a publicly available tool[1], "aimed to simulate astronomical images obtained with any (given) telescope and instrument combination"[20]. It allows modeling an astronomical target either as a black body radiator or from a file containing the spectral flux densities. Furthermore, the background radiation, atmospheric extinction as well as mirror and instrument efficiencies can be modeled with AETC. However, only an imaging detector is available.
Thermal emissions of the telescope components as well as multiple optical surfaces cannot be taken into account directly but have to be included using a workaround. As AETC provides an interface to include arbitrary sky background emission, the thermal emission of the telescope can be included as sky background emission. However, the telescope's thermal emission has to be properly scaled to take the subsequent atmospheric extinction into account.

[1] `http://aetc.oapd.inaf.it`

The effect of multiple imperfect optical surfaces can be taken into account using a second workaround: the reflection and transmittance coefficients of all surfaces can be multiplied to give the overall system coefficient which can be inputted to AETC as a single mirror reflection coefficient.
These workarounds allowed AETC to be used as a reference for the verification of many requirements. All targets and optical components as well as the imaging detector could be verified with this exposure time calculator. Beyond that, the airy PSF implementation could be compared to AETC and verified.

5.1.2. SOFIA SITE Verification

SOFIA Instrument Time Estimator (SITE) is "a web-based tool that provides total integration time or S/N for a given instrument, filter(s), source type (point, extended, emission line), and water vapor overburden"[7]. It allows simulating observations with all instruments of SOFIA including the German Receiver for Astronomy at Terahertz Frequencies (GREAT). GREAT is a heterodyne spectrometer developed and maintained by a consortium of German research institutes. Using SITE to simulate observations with GREAT allowed to verify all requirements of the heterodyne instrument provided by ESBO-ETC .

5.1.3. Verification with Hand Calculations

The effect of pointing jitter is not considered transparently in any other exposure time calculator. For this reason, hand calculations were used for the verification of this requirement. ESBO-ETC models pointing jitter by a convolution of the PSF with a Gaussian bell curve with a given standard deviation (eq. (2.34). An analytical solution with manual calculations of this convolution is possible if the PSF is approximated by another Gaussian bell curve. This has been done for the airy disk representation of the PSF.

5.1.4. Verification with Zemax

To verify the correct implementation of the Zemax PSF in ESBO-ETC , the photometric aperture's diameter as calculated by ESBO-ETC was compared to the result of a computation carried out in Zemax for a given percentage of encircled energy.

5.1.5. Verification with ESBO-ETC

The implementation of the FITS representation of the airy disk was verified by comparing the photometric aperture's diameter for the verified airy disk implementation with the corresponding diameter for a FITS-file containing the same PSF.

5.2. Verification Results

All requirements defined in [5] have been tested during the software verification. The result of each verification has proven to be within a reasonable acceptance range. All details on the results are available in [19]. To make the verification reproducible, all used configuration files and data files are delivered with this thesis.

6. Analysis of two Scientific Applications

After the software has been successfully verified against the software requirements, a validation shall assess the applicability of the new software for its use cases. For this reason, ESBO-ETC is applied in two science cases, one with an imaging detector and one with a heterodyne detector. The scientific question, the setup including all sources and the results of both scenarios are described in the following sections.

6.1. SOFIA FORECAST - Influence of Mirror Coatings on Integration Time

The Stratospheric Observatory for Infrared Astronomy (SOFIA) is a flying observatory for the infrared regime aboard a Boeing 747-SP. Its telescope is designed as a cassegrain telescope in a nasemyth configuration with a usable primary mirror diameter of 2.5 m. The light collected by the parabolic primary mirror is reflected by an opposing hyperbolic secondary mirror onto the tertiary mirror that deflects the collected light through the nasemyth tube into the science instrument[21]. SOFIA allows to change the science instrument during the ground-time and offers several instruments for different wavelength ranges and resolutions.
SOFIA's telescope, which was built by a German consortium lead by the DLR and is now maintained by the DSI, is under continuous development. Among other things, the secondary mirror (assembly) is of particular interest, as this comparably small component ($\varnothing$350 mm) with its chop-nod-mechanism is not only the most integrated component of the telescope but can also be exchanged relatively easy. By replacing the aluminum coating of the mirror with a better coating like gold, not only the loss along the optical path can be reduced but also the thermal emission of the mirror. As observations in the FIR suffer from large background emissions of the telescopic components, this can significantly improve the observations.
To assess the influence of the secondary mirror coating on the integration time, an observation of the star *Pleione* (BU Tau / 28 Tau) in the Pleiades (Messier 45) with the instrument FORECAST shall be simulated using ESBO-ETC . First of all, a simulation with the current telescope parameters shall be carried out and compared to the SITE. In a second step, the simulation shall be carried out for the improved coating.
The Faint Object Infrared Camera for the SOFIA Telescope (FORECAST) is one of SOFIA's science instruments and offers two 256×256 pixel arrays for a wavelength range of $5 - 25\,\mu$m (SWC) and $25 - 40\,\mu$m (LWC). A suite of different grisms allows FORECAST to be used not only as an imaging detector but also as a spectroscopic detector. In the imaging mode, each array exhibits a quadratic FOV of $3.2' \times 3.2'$ with a plate scale of $0.75''$ per pixel. SOFIA's telescope beam enters the instrument at the science instrument flange through a dewar window onto the imaging collimator mirror and then over a fold mirror into the liquid helium cooled cryostat. The scientist can choose between a dichroic mirror as beamsplitter for simultaneous operation of both arrays or an empty slot/mirror to use only a single array for the observation.

After passing through FORECAST's filter array for each channel, two additional fold mirrors redirect the beam onto the detector[22].

6.1.1. Simulation Parameters

Parameter	Value	Unit	Source
Source temperature	12106	K	[23]
Source apparent magnitude	5.19	mag	[23]
Cosmic background temperature	2.725	K	[24]
Observation altitude	41,000	ft	[25]
Elevation angle	40	degree	[25]
Water vapor overburden	7.1	µm	[25]
Sky temperature	240	K	[25]
Primary mirror diameter	2.5	m	[25]
Primary mirror reflectivity	0.965	-	[25]
Primary mirror temperature	240	K	[25]
Secondary mirror reflectivity	0.965	-	[25]
Secondary mirror temperature	240	K	[25]
Tertiary mirror reflectivity	0.965	-	[25]
Tertiary mirror temperature	240	K	[25]
Dewar window transmittance	0.88	-	[25]
Dewar window temperature	290	K	[25]
FORECAST mirror reflectivity	0.97	-	[25]
FORECAST mirror temperature	4	K	[22]
Collimator mirror temperature	77	K	[22]
Filter central wavelength	19.712	µm	[25]
Filter bandwidth	5.506	µm	[25]
Filter temperature	4	K	[22]
Read noise	250	e^-	[22]
Dark current	$7 \cdot 10^4$	$e^- s^{-1}$	[22]
Quantum efficiency			[22]
Excess noise	2.5	-	[25]
Pixel size	50	µm	[25]
Pixel FOV	0.75	arcsec	[25]
Well depth	$1.6 \cdot 10^6$	e^-	[22]
Photometric aperture diameter	5.3	arcsec	[25]

Table 6.1.: Parameters and the according sources used for the simulation of an observation of the star *Pleione* with SOFIA/FORECAST. Note that the dark current and read noise are given for the full array and not per pixel.

Each component of the telescope and the detector has to be modeled in ESBO-ETC to conduct a realistic simulation. Therefore the characteristic properties of each component have to be collected. The parameters used for the simulation are listed in table 6.1 and the corresponding ESBO-ETC configuration file is shown in appendix F.1 and executable on Code Ocean[26].

The working focal number N of the setup can be calculated from the pixel size l, the pixel FOV β and the aperture diameter D_{ap} as

$$N = \frac{f}{D_{ap}} = \frac{l}{2 \cdot D_{ap} \cdot \tan(\beta)} = 5.50. \tag{6.1}$$

The mean radiant flux density of the source required by SITE can be obtained by integrating equation (2.9) over the filter range λ_0 – λ_1 and dividing by the filter's bandwidth $\Delta\lambda$

$$\overline{E_\lambda} = \frac{1}{\lambda_1 - \lambda_0} \cdot \int_{\lambda_0}^{\lambda_1} E_{\lambda,Planck}\, \mathrm{d}\lambda = 8.7397 \cdot 10^{-16}\, \frac{\mathrm{W}}{\mathrm{m}^2 \cdot \mu\mathrm{m}}. \tag{6.2}$$

FORECAST distinguishes between the responsive quantum efficiency and the detective quantum efficiency (see [22] for further information). The detective quantum efficiency was used for the simulations.

6.1.2. Simulation Results

The accuracy of an ESBO-ETC simulation with the parameters given in table 6.1 has to be determined before the influence of different mirror coatings on the required integration time can be assessed. Therefore, the same simulation is conducted with ESBO-ETC and SITE to gain an insight into ESBO-ETC 's accuracy. Different mirror coatings are simulated in a second step to assess their influence.

Comparison of ESBO-ETC and SITE

A simulation was conducted with SITE and ESBO-ETC using the parameters given above to compute the integration time required to reach a SNR of 4. Even though the observation clearly suffers from high background radiation and requires chopping and nodding, the simulation is carried out for a single exposure without chopping. This said the readout noise of the detector is only considered once and not for each chop frame. This proceeding is in accordance to SITE.
The results of both computations are listed in table 6.2. The result of ESBO-ETC is very close to SITE and verifies therefore once more the implementation of all equations. The slight difference of 16 s can be attributed to a possible differing calculation of the number of pixels contained in the virtual photometric aperture.

Tool	SNR	Integration Time
SITE	4.0	252 s
ESBO-ETC	4.0	268 s

Table 6.2.: Result of the first simulation to compare ESBO-ETC to SITE.

Analysis of Improved Mirror Reflectivity

To analyze the influence of different mirror coatings on the required integration time for an observation of Pleione a gold coating is assumed for the secondary mirror and for both

the primary mirror and the secondary mirror. The reflectivity coefficients used are listed in table 6.3.

Parameter	Value	Unit
Primary mirror reflectivity	0.99	-
Secondary mirror reflectivity	0.99	-

Table 6.3.: Parameters and the according sources used for the simulation of an observation of the star *Pleione* with SOFIA FORECAST. Note that the dark current and read noise are given for the full array and not per pixel.

Conducting the ESBO-ETC simulation a second time with the improved parameters leads to the results in table 6.4. The results clearly show a reduction of approximately 10% of the required integration time for each gold-coated mirror. This result means, that more observations can be conducted per science flight than before. However, the number of observations does not scale linearly to the integration time due to calibration and slewing overheads. A re-coating of the primary mirror is furthermore unrealistic due to the high risk of damage during removal and coating. The secondary mirror, on the other hand, can be easily removed and, in case of damage, manufactured a second time. This said a new, more efficient coating of the secondary mirror is advisable.

Improvement	SNR	Integration Time	Improvement
Gold coated M2	4.0	239 s	10.8%
Gold coated M1 + M2	4.0	212 s	20.9%

Table 6.4.: Result of the second simulation with improved mirror coatings.

6.2. ESBO Heterodyne Instrument - HCl^+ absorption towards W31C

"Hydrides play a central role in interstellar chemistry, both as significant reservoirs of heavy elements and as critical intermediaries in the pathways leading to more complex molecules"[27]. "Because the chemical pathways leading to the formation of interstellar hydrides are fairly simple, the analysis of the observed abundances is straightforward, and provides key information about the physical and chemical conditions within the environments in which hydrides are found"[28].
The PRISMAS program (Probing InterStellar Molecules with Absorption line Studies) was conducted on the satellite observatory Herschel using the Heterodyne Instrument for the Far Infrared (HIFI) to observe around 20 molecules towards 8 sources. Because of the molecule's small moments of inertia, their rotational transitions lie at high frequencies that are often inaccessible for ground-based telescopes[28] which is the reason why Herschel was chosen. Due to Herschel's retirement in 2013, no space-based observatory is currently available for observations in this frequency range.
To overcome this issue, the observational capabilities of a heterodyne instrument like GREAT on ESBO will be assessed in this section. For this reason, the observation of the HCl^+ absorption line (1444.2 GHz / 207.6 µm) in the interstellar medium towards W31C, as conducted in the PRISMAS program[28], will be modeled with ESBO-ETC . Westerhout 31C or G10.6-0.4 located in the constellation of Sagittarius is a complex of star formation regions in the Milky Way. First of all, an ESBO-ETC simulation is carried out for Herschel HIFI to gain insight into the accuracy of the chosen model. The same observation will be subsequently simulated in a second step using the ESBO telescope and the instrument parameters of GREAT.

6.2.1. Simulation Parameters

Again, all parameters of the simulated components have to be collected in advance of the simulations. A complete list of all used parameters is given in table 6.5 for Herschel and 6.6 for ESBO.
As a star-forming region, W31C is clearly a largely extended source that can be associated with a brightness temperature. The Herschel Science Archive (HSA) gives a DSB temperature of $T_{A,DSB}^* = 10.5\,\mathrm{K}$ for the observation of W31C[1]. Due to an approximately equal distribution between the USB and the LSB, the DSB temperature appears as twice its real value[15]. Therefore $T_{A,SSB}^* = 5.25\,\mathrm{K}$. Even though the quantity $T_A{}^*$ is calibrated to exclude several instrumental effects, it is still a HIFI specific quantity and yet does not take any telescopic losses of Herschel into account. The main beam efficiency $\eta_{mb} = 0.58$ takes these losses into account which is the reason why the antenna temperature has to be converted into the main beam temperature[29] using equation (2.44). This yields $T_{mb} = 8.7\,\mathrm{K}$ which is equal to W31C's brightness temperature and a Planck black body temperature of $T_B = 31.6\,\mathrm{K}$.
A comparable temperature arises for an observation[2] of W31C using the Photodetector Array Camera and Spectrometer (PACS) instrument of Herschel that was also conducted in the scope of PRISMAS. The Herschel Science Archive provides a spectral flux density value from 70 µm up to 190 µm which can be extrapolated to the line wavelength of 207.6 µm which

[1] Observation ID: 1342206601
[2] Observation ID: 1342217945

gives a spectral flux density of approximately $E_\nu = 1200\ \frac{\mathrm{Jy}}{\mathrm{spaxel}}$. The PACS handbook [30] gives a spaxel size of 9.4" × 9.4" resulting in a spectral radiance of $L_{\Omega,\nu} = 5.77\ \frac{\mathrm{W}}{\mathrm{m^2 \cdot Hz \cdot sr}}$ and a brightness temperature of $T_B = 32.0\,\mathrm{K}$.
The Herschel Science Archive gives a DSB line antenna temperature of $T_{A,DSB}^{*} = 9.5\,\mathrm{K}$ which equals a single sideband antenna temperature of $T_{A,DSB}^{*} = 4.25\,\mathrm{K}$ as the line is only present in the upper sideband. This leads to a line extinction of 19% or an equivalent transmission of 81%.
As mentioned before, the main beam efficiency η_{mb} takes the telescopic losses of Herschel into account. As ESBO-ETC simulates the telescope components and their losses separately, the main beam efficiency has to be normalized for the telescopic losses by dividing the main beam efficiency by the telescope efficiency. This yields $\eta_{mb} = 0.64$.
W31C is both located in the Milky Way and close to the ecliptic which is the reason why the galactic cirrus and zodiacal background radiation is considered in the simulations.
The corresponding configuration file for the simulation is given in appendix F.2 and executable on Code Ocean[31].

Parameter	Value	Unit	Source
Source temperature	31.6	K	HSA
HCL^+ absorption	0.19	-	HSA
Zodiacal light temperature	274	K	[32]
Galactic cirrus temperature	20	K	[33]
Cosmic background temperature	2.725	K	[24]
Primary mirror diameter	3.5	m	[34]
Primary mirror reflectivity	0.95	-	[34]
Primary mirror temperature	88	K	HSA meta info
Secondary mirror reflectivity	0.95	-	
Secondary mirror temperature	84	K	HSA meta info
Aperture efficiency	0.6	-	[15]
Main beam efficiency	0.64	-	[15]
Receiver temperature	1050	K	[15]
Antenna forward efficiency	0.96	-	[15]
Local oscillator frequency	1441.5	GHz	HSA meta info
Resolution	1.1	MHz	[28]
Integration time	404.48	s	HSA meta info

Table 6.5.: Parameters and the according sources used for the simulation of an observation of the HCL^+ absorption towards W31C with Herschel/HIFI. Note that the secondary mirror reflectivity was assumed to be the same as the primary mirror.

6.2.2. Simulation Results

Before a heterodyne instrument on ESBO can be simulated, the modeled target and background emissions have to be validated against the performed observation with Herschel/HIFI. Both results are shown and explained in this section.

Parameter	Value	Unit	Source
Observation altitude	40.0	km	[1]
Zenith angle	0	degree	
Water vapor overburden	7.1	µm	[25]
Sky temperature	265	K	[35]
Primary mirror diameter	5	m	[1]
ESBO mirror reflectivity	0.95	-	P. Maier
Primary mirror temperature	-15	°C	[36]
M2, M3, M4, M5 temperature	-20	°C	[36]
Aperture efficiency	0.55	-	[7]
Main beam efficiency	0.67	-	[7]
Receiver temperature	1000	K	[7]
Antenna forward efficiency	0.97	-	[7]

Table 6.6.: Alternative and supplementary parameters and the according sources used for the simulation of an observation of the HCL^+ absorption towards W31C with ESBO. Note that the computation of the atmospheric transmission using ATRAN was performed for the maximum allowed altitude of 29.1 km and not for 40 km.

Comparison of ESBO-ETC and HSA

The Herschel Science Archive gives an RMS noise temperature of $\Delta T_{rms} = 0.222\,\mathrm{K}$ for the HIFI observation which yields an SNR of 42.7 with the DSB line antenna temperature given above. The result of the simulation with ESBO-ETC is given in table 6.7 which shows a good agreement with the HSA data. This means, that the used models for the target and background emissions are valid and can be used for the simulation of a heterodyne instrument aboard of ESBO.

Tool	SNR	Integration Time
HSA	42.7	404.48 s
ESBO-ETC	42.7	418.9 s

Table 6.7.: Result of the first simulation to compare ESBO-ETC to the observation results in HSA.

Analysis of a Heterodyne Instrument on ESBO

The capabilities of a heterodyne instrument on ESBO with similar properties like GREAT has been assessed by simulating the observation of HCL^+ towards W31C with ESBO-ETC . The result of this simulation is listed in table 6.8 which shows a lower performance capability of ESBO as compared to Herschel. The main reason for this is the significantly higher mirror temperature of ESBO (90 K as compared to 250 K). Furthermore, ESBO makes use of three additional mirrors to decouple the instruments from the telescope elevation movement which reduce the collected radiation. The used configuration file is shown in appendix F.2 and the simulation is executable on Code Ocean[37].

Nevertheless, this result is very promising because it clearly shows that ESBO is able to perform observations in this wavelength regime within a realistic integration time. As the reflectivity coefficients of all mirrors are only worst-case estimates by P. Maier, the real performance of a heterodyne instrument on ESBO can be better than simulated.

SNR	Integration Time	Deviation
42.7	533.1 s	+31.8%

Table 6.8.: Result of the second simulation with improved mirror coatings.

7. Conclusion

To conclude, the development of the modular exposure time calculator was successful. All equations required for the computations have been derived and collected. A state-of-the-art software architecture was developed ensuring the modularity of the software. Furthermore, the chosen software architecture enables easy future updates and extensions of the ETC.
A documentation provides information on the installation and usage of ESBO-ETC for all users of the software. Additional information is given for future developers that help to understand the structure of the code and how extensions can be developed.
A careful verification of the software was successfully conducted ensuring the fulfillment of all requirements. Subsequently, ESBO-ETC was validated by means of two science cases. A simulation of an observation with SOFIA/FORECAST was conducted on the one hand and compared to the SOFIA exposure time calculator SITE. The influence of different coatings of the primary and secondary mirrors on the required integration time was analyzed using ESBO-ETC . Furthermore, the performance capabilities of a heterodyne instrument like GREAT on ESBO were assessed with a simulated observation of the HCL^+ absorption towards W31C that was previously performed by Herschel/HIFI. It has been shown that even though ESBO cannot achieve the same sensitivity as Herschel, it promises a great benefit for the scientific community in this wavelength regime that is inaccessible from the ground.
The thesis is delivered with a set of documents for the software requirements[5] and verification[19] as well as all input files for the verification and validation cases. The source code of ESBO-ETC 's first version is also included in the delivered bundle.

7.1. Future Upgrades & Extensions

ESBO-ETC is designed to make future updates and extensions as easy as possible. Even though the first version is not only a proof of concept but already a mature software that can be used for many use cases, it shall pave the road to an extensive exposure time calculator that is widely used.
A milestone to come to this state is the development of a graphical user interface (GUI) that allows the users of ESBO-ETC to create the required configuration in a visually appealing environment rather than in an XML-editor. This GUI should enable the user to not only create a configuration file but also to conduct a simulation and analyze the results. Like any modern application, ESBO-ETC 's GUI should be served as a web application that can be accessed and used by people all around the world over the internet and does not require a complicated installation process. An architecture for such an application could consist of a python *flask*-server[1] in the back-end that imports the ESBO-ETC python module and a *vue.js*[2] front-end. The framework *Rete.js*[3] enables visual programming by dragging and connecting different

[1] `https://flask.palletsprojects.com`
[2] `https://vuejs.org`
[3] `https://rete.js.org/`

types of nodes. Each component offered by ESBO-ETC could be represented by a node and connected to other nodes to model the radiative transfer path.
Another possibility to create a graphical user interface for editing the configuration file are XML stylesheets that enable most web browsers to display XML in a predefined layout. This layout may provide select inputs and buttons to alter and add the different components.
Besides the development of a GUI, some minor improvements can be added to the existing software. Considering the polarization of the emitted radiation of all components would allow a more precise simulation of the polarization-sensitive heterodyne instrument. This is for example useful for the simulation magnetic field observations that are made by measuring the polarization of the radiation strayed by dust grains in the magnetic field.
Furthermore, the antenna pattern could be included in the computations of the heterodyne instrument like it is the case for the PSF for the imaging detector. This improvement might enhance the usability of ESBO-ETC for the developers of heterodyne detectors.
Even though ESBO-ETC natively supports the computation of the atmospheric transmission using an online version of ATRAN and parses output files of other tools, the implementation of an interface to other atmospheric transmission calculators would be beneficial. This is due to the fact that ATRAN is limited to a maximum observation altitude of 29.1 km which might be problematic for balloon observatories in certain wavelength regimes. To overcome this issue an interface for the transmission calculator *am* could be implemented, as *am* has no such limitation.
Lastly, extensive plotting functionalities could be added to ESBO-ETC to enable the creation of plots at any point in the optical path. This is possible by introducing a new optical component that just forwards the incoming radiation to the calling component and simultaneously plots these quantities. Additional plotting functionalities could be added to the detector components to create plots of the detected signals. Both additions not only improve the result output but also enable the user to track issues during the development of a configuration file.

A. Standard photometric systems

The standard photometric systems used for flux normalization of the black body target in section 2.2 are tabulated below.

Spectral band	Central wavelength λ_0 in μm	Bandwidth $\Delta\lambda_0$ in μm	E_λ in $\frac{\mathrm{W}}{\mathrm{m}^2 \cdot \mathrm{um}}$
U	0.365	0.068	$4.27 \cdot 10^{-8}$
B	0.44	0.098	$6.61 \cdot 10^{-8}$
V	0.55	0.089	$3.64 \cdot 10^{-8}$
R	0.7	0.22	$1.74 \cdot 10^{-8}$
I	0.9	0.24	$8.32 \cdot 10^{-9}$
J	1.25	0.3	$3.18 \cdot 10^{-9}$
H	1.65	0.4	$1.18 \cdot 10^{-9}$
K	2.2	0.6	$4.17 \cdot 10^{-0}$
L	3.6	1.2	$6.23 \cdot 10^{-11}$
M	4.8	0.8	$2.07 \cdot 10^{-11}$
N	10.2	2.5	$1.23 \cdot 10^{-12}$

Table A.1.: Standard photometric systems
Prominent spectral bands, their central wavelength and bandwidth and the corresponding spectral flux density of a $0\ ^{\mathrm{mag}}$ star are tabulated using the Vega magnitude system[38].

B. Derivation of ExoSim

ExoSim does not provide a public paper about the physics used for the calculations. However, the source code is publicly available under the reproducible research software license. The following derivation is based on lines 88 to 96 of ExoSim's source code [1] and tries to show the parallels to the derivation used in this thesis.
The flux Φ_ν received from an extended source with spectral radiance $L_{\Omega,\nu}$ can be calculated as

$$\Phi_\nu = L_{\Omega,\nu} \cdot \Omega_{ap} \cdot l^2. \tag{B.1}$$

By definition, the telescope's FOV solid angle Ω_{ap} is given by

$$\Omega_{ap} = \frac{A}{r^2} \tag{B.2}$$

where A is the spherical surface area and r is the radius of the considered sphere as shown in figure B.1.

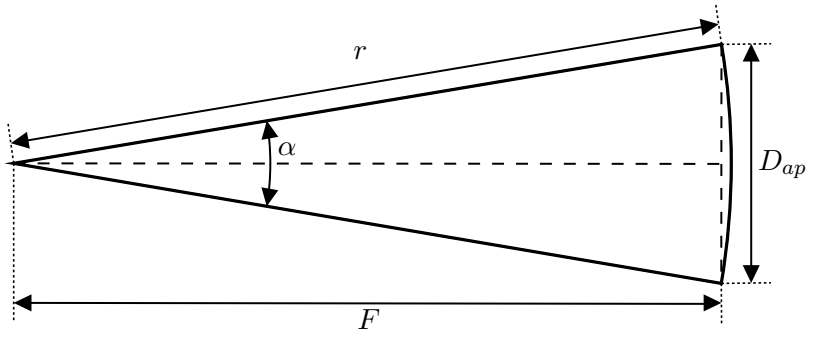


Figure B.1.: Sketch for the calculation of the telescope's FOV solid angle Ω_{ap}

The spherical surface area A of a cone with an apex angle α is given by a formula collection as

$$A = 2\pi \cdot r^2 \cdot \left(1 - \cos\left(\frac{\alpha}{2}\right)\right). \tag{B.3}$$

Using trigonometric functions, the cone's apex angle α is given by

$$\frac{\alpha}{2} = \arccos\left(\frac{f}{\sqrt{f^2 + \left(\frac{D_{ap}}{2}\right)^2}}\right) = \arccos\left(\frac{N}{\sqrt{N^2 + \frac{1}{4}}}\right) \tag{B.4}$$

[1] https://github.com/ExoSim/ExoSimPublic/blob/d690dcd2c89325b0c617b8de9b66177ced57b016/exosim/modules/instrument.py#L88-L96

as shown in sketch B.1. Plugging equation (B.4 into B.3) leads to

$$A = 2\pi \cdot r^2 \cdot \left(1 - \frac{N}{\sqrt{N^2 + \frac{1}{4}}}\right) = 2\pi \cdot r^2 \cdot \frac{N^2 + \frac{1}{4} - N \cdot \sqrt{N^2 + \frac{1}{4}}}{N^2 + \frac{1}{4}} \tag{B.5}$$

which can be simplified to

$$A = \pi \cdot r^2 \cdot \frac{1}{4N^2 + 1}. \tag{B.6}$$

The error ε of this approximation is given by

$$\varepsilon = \frac{2 \cdot \frac{N^2 + \frac{1}{4} - N \cdot \sqrt{N^2 + \frac{1}{4}}}{N^2 + \frac{1}{4}} - \frac{1}{4N^2 + 1}}{2 \cdot \frac{N^2 + \frac{1}{4} - N \cdot \sqrt{N^2 + \frac{1}{4}}}{N^2 + \frac{1}{4}}} = \frac{2N^2 + \frac{1}{4} - 2N \cdot \sqrt{N^2 + \frac{1}{4}}}{N^2 + \frac{1}{4} - N \cdot \sqrt{N^2 + \frac{1}{4}}} \tag{B.7}$$

which converges to zero for $N \to \infty$. The error is bounded by 0.2481% for $N > 5$. This approximated equation in turn can be plugged together with (B.2 into eq. B.1) resulting in

$$\Phi_\nu = L_{\Omega,\nu} \cdot l^2 \cdot \frac{\pi}{4N^2 + 1}. \tag{B.8}$$

This final equation is equivalent to equation (2.26) and therefore confirms the approach used in this thesis.

C. Approximation Error for the Obstructed Extended Source

The approximations used to get from equation (2.30 to 2.31 in section 2.4.1 are examined in this chapter. As the error of these approximations depends on both the value of focal number N as well as the value of the obstruction ration o and the ratio of the radiances $L_{\Omega,\nu,1,obs}$ and $L_{\Omega,\nu,1}$, figure C.1) shows the relative approximation error ε with respect to these values. The relative approximation error is calculated by

$$\varepsilon = \frac{\Phi_{\nu,1-2,\text{simplified}} - \Phi_{\nu,1-2}}{\Phi_{\nu,1-2}} = \frac{o\,(1-o)\cdot(L_{\Omega,\nu,1} - L_{\Omega,\nu,1,obs})}{L_{\Omega,\nu,1,obs}\cdot o\,(4N^2+1) + L_{\Omega,\nu,1}\cdot 4N^2\,(1-o)} \tag{C.1}$$

from equation (2.30 and 2.31).

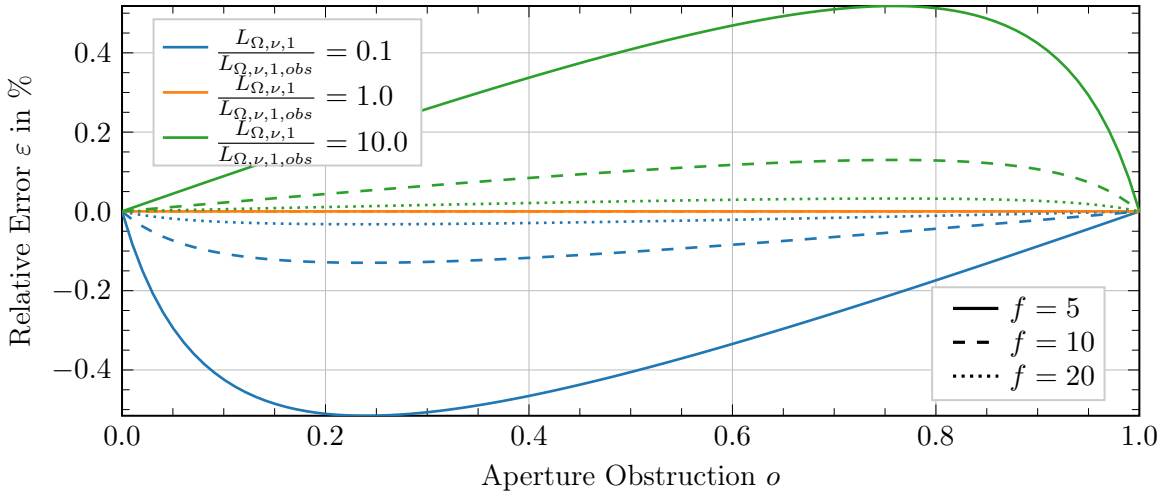


Figure C.1.: Relative approximation error in the derivation of the received spectral flux of an obstructed extended source for different obstruction ratios o
The line color encodes different ratios of the source radiance to the obstructing component's radiance and different focal numbers N are indicated by the line pattern.

Equation (C.1 as well as figure C.1 clearly show, that the error tends to zero for N) approaching infinity

$$\lim_{N\to\infty} \varepsilon = 0.$$

However, telescopes in reality typically have a focal number N in the order of 10^1–10^2. Therefore the contribution of the obstruction ratio and the component's radiances has to be further assessed.

As indicated in figure (C.1, the error for unobstructed radiation transportation ($o = 0$) is zero whereas the error stays limited for $0 \leq o) \leq 1$ with the maximal error

$$\max(\varepsilon) \approx \frac{\sqrt{\frac{L_{\Omega,\nu,1}}{L_{\Omega,\nu,1,obs}}} - 1}{4N^2(\sqrt{\frac{L_{\Omega,\nu,1}}{L_{\Omega,\nu,1,obs}}} + 1) + 1} \quad \text{at} \quad o \approx 1 - \frac{1}{\sqrt{\frac{L_{\Omega,\nu,1}}{L_{\Omega,\nu,1,obs}}} + 1}.$$

For an equal spectral radiance of both the obstructing component and the incoming radiation, the approximation error is zero and increases for differing values as shown in figure C.1.
By limiting the focal number to $N \geq 5$, the relative approximation error is bounded by 1% for all obstruction ratios and all radiance ratios. This error is reasonable and therefore justifies the approximations.

D. Class Diagram

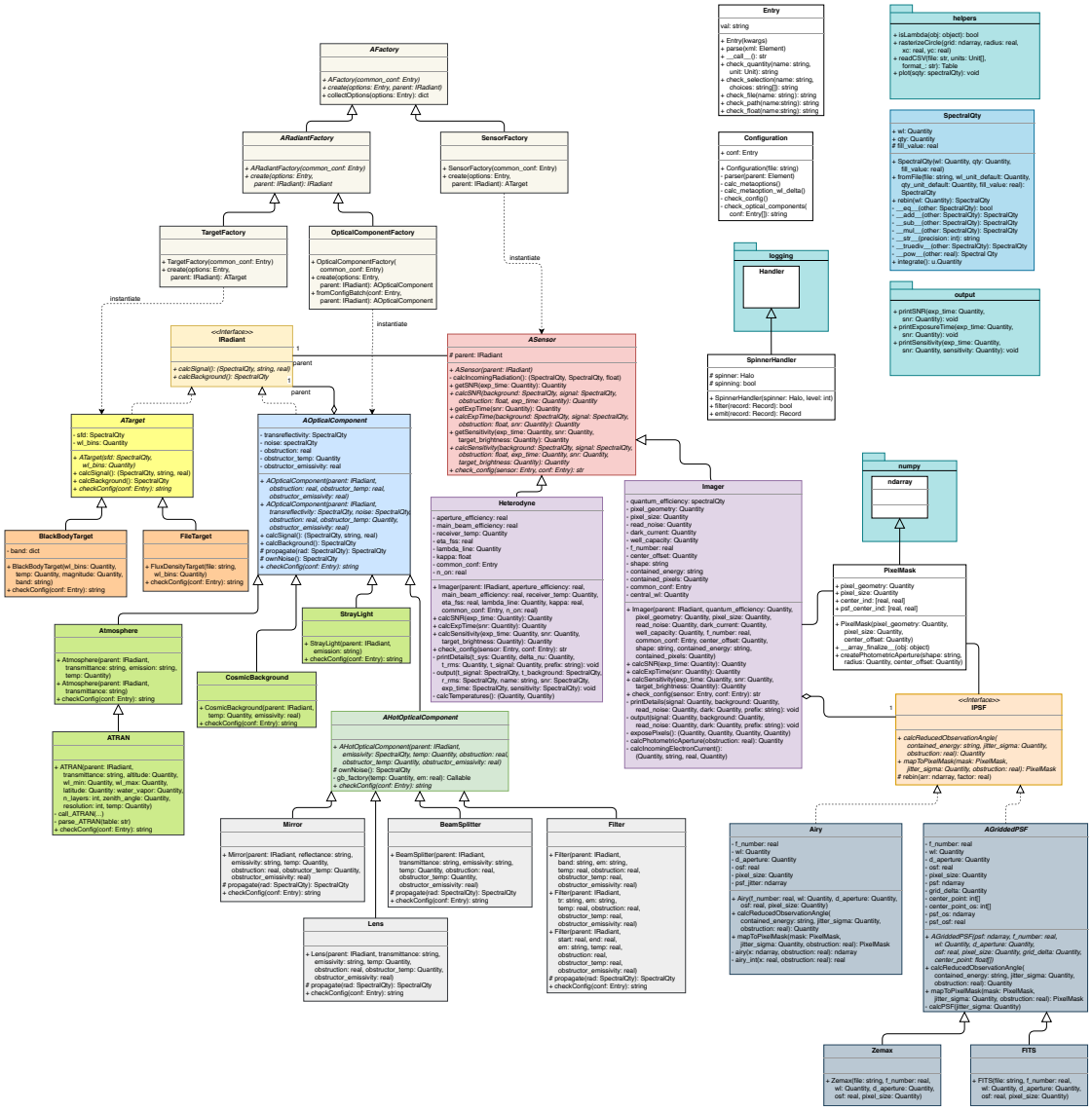


Figure D.1.: Complete class diagram of ESBO-ETC
The diagram can be found in full resolution in the documentation of ESBO-ETC `https://esbo-ds.irs.uni-stuttgart.de/esboetcdocs/developer/developer_documentation.html#software-architecture`.

E. Verification Results

The description and results of all verification cases as explained in section 5 are listed in table E.1

#	Requirement	Method	Result	Deviation
3.2a)	Blackbody target	AETC	$SNR_{AETC} = 53.77$ $SNR_{ESBO-ETC} = 54.19$	0.78%
3.2b)	File target	ESBO-ETC	$(SNR_{AETC} = 53.78)$ $SNR_{ESBO-ETC} = 54.19$	0.76%
3.2	Extended target	SITE	$SNR_{SITE} = 4.00$ $SNR_{ESBO-ETC} = 3.81$	4.75%
3.3	Atmosphere	AETC	$SNR_{AETC} = 52.78$ $SNR_{ESBO-ETC} = 53.09$	0.59%
3.4	Cosmic background	AETC	$SNR_{AETC} = 34.56$ $SNR_{ESBO-ETC} = 33.55$	2.92%
3.5	Stray Light	AETC	$SNR_{AETC} = 53.17$ $SNR_{ESBO-ETC} = 53.50$	0.62%
3.7a)	Mirror	AETC	$SNR_{AETC} = 50.85$ $SNR_{ESBO-ETC} = 51.23$	0.75%
3.7b)	Lens	AETC	$SNR_{AETC} = 50.85$ $SNR_{ESBO-ETC} = 51.23$	0.75%
3.7c)	Beam splitter	AETC	$SNR_{AETC} = 50.85$ $SNR_{ESBO-ETC} = 51.23$	0.75%
3.7d)1)	Filter from band	AETC	$SNR_{AETC} = 46.51$ $SNR_{ESBO-ETC} = 46.86$	0.75%
3.7d)2)	Filter from file	AETC	$SNR_{AETC} = 41.32$ $SNR_{ESBO-ETC} = 41.63$	0.75%
3.7d)3)	Custom filter	AETC	$SNR_{AETC} = 43.79$ $SNR_{ESBO-ETC} = 44.13$	0.78%
3.7	Thermal emission	AETC	$SNR_{AETC} = 36.79$ $SNR_{ESBO-ETC} = 35.74$	2.85%
3.8	Obstruction	AETC	$SNR_{AETC} = 49.28$ $SNR_{ESBO-ETC} = 49.68$	0.81%
3.8	Thermal emission	AETC	$SNR_{AETC} = 36.02$ $SNR_{ESBO-ETC} = 35.99$	0.08%
3.10a)	Imager quantum efficiency	AETC	$SNR_{AETC} = 29.94$ $SNR_{ESBO-ETC} = 30.17$	0.77%
3.10a)	Imager read out noise	AETC	$SNR_{AETC} = 51.14$ $SNR_{ESBO-ETC} = 51.08$	0.12%
3.10a)	Imager dark current	Manual Calculation	$SNR_{AETC} = 53.48$ $SNR_{ESBO-ETC} = 53.84$	0.67%

Table E.1 continued from previous page

#	Requirement	Method	Result	Deviation
3.10a)	Imager photometric aperture size	Manual Calculation	$r_{\text{phot}} = 7.17$ pixels $r_{\text{phot,ESBO-ETC}} = 7.18$ pixels	0.14%
3.10a)	Imager photometric aperture shape	ESBO-ETC	$\text{SNR}_{\text{circular}} = 55.11$ $\text{SNR}_{\text{square}} = 55.62$	-
3.10a)	Imager photometric aperture contained pixels	ESBO-ETC	$\text{SNR}_{\text{EE}} = 54.76$ $\text{SNR}_{\text{fixed}} = 54.76$	0.00%
3.10a)	Imager: centroid shift	ESBO-ETC	$\text{SNR}_{\text{center}} = 14.59$ $\text{SNR}_{\text{edge}} = 13.13$ Result is symmetric	-
3.10b)	Heterodyne	SITE	$t_{\text{exp,SITE}} = 1362.0$ s $t_{\text{exp,ESBO-ETC}} = 1362.4$ s	0.03%
3.10b)	Heterodyne mapping	SITE	$t_{\text{exp,SITE}} = 448.2$ s $t_{\text{exp,ESBO-ETC}} = 448.3$ s	0.02%
4.1a)	SNR	AETC	$\text{SNR}_{\text{AETC}} = 11.67$ $\text{SNR}_{\text{ESBO-ETC}} = 11.45$	1.86%
4.1b)	Exposure Time	ESBO-ETC	$t_{\text{exp,initial}} = 0.1$ s $glsSNR_{\text{ESBO-ETC}} = 11.45$ $t_{\text{exp,ESBO-ETC}} = 0.1$ s	0.00%
4.1c)	Sensitivity	ESBO-ETC	$\text{SNR}_{\text{initial}} = 5.00$ Sensitivity = 10.99 mag $\text{SNR}_{\text{ESBO-ETC}} = 5.00$	0.00%
5.1	Jitter Airy	Manual Calculation	$r_{\text{phot,jit}} = 6.95$ pixels $r_{\text{phot,,jit,ESBO-ETC}} = 7.00$ pixels	0.72%
5.1	Jitter GriddedPSF	ESBO-ETC	$r_{\text{phot,jit,airy}} = 12.40$ pixels $r_{\text{phot,jit,FITS}} = 12.38$ pixels	0.16%
5.2a)	Airy PSF	AETC	See 4.1a)	-
5.2b)	Zemax PSF	Zemax	$r_{\text{phot}} = 0.97$ pixels $r_{\text{phot,ESBO-ETC}} = 0.95$ pixels	2.06%
5.2c)	FITS PSF	ESBO-ETC	$r_{\text{phot,airy}} = 7.18$ pixels $r_{\text{phot,FITS}} = 6.89$ pixels	4.04%
-	Overall	AETC	See 4.1a)	-

Table E.1.: All verification cases are listed with their corresponding requirement number[5], a short description of the requirement, the used verification method and the results of the verification.

F. Science Case Configuration Files

The configuration files used in the science cases as described in chapter 6 are listed below. The required input files are delivered with this thesis.

F.1. FORECAST Configuration File

Listing 1 shows the configuration file used for the simulation of an observation of Pleione with SOFIA FORECAST as described in section 6.1.

```
<root>
    <common>
        <wl_min val="1.71 um"/>
        <wl_max val="37.71 um"/>
        <wl_delta val="10 nm"/>
        <d_aperture val="2.5 m"/>
        <psf type="Airy" osf="10"/>
        <output path="output" format="fits"/>
        <snr val="4"/>
    </common>

    <astroscene>
        <target type="BlackBodyTarget" temp="12106 K" mag="5.19 mag" band="N"
        ↪  comment="Pleione"/>
        <optical_component type="CosmicBackground" temp="2.725 K"
        ↪  comment="Cosmic Background"/>
        <optical_component type="ATRAN" altitude="41000 ft" wl_min="15.71 um"
        ↪  wl_max="23.71 um" latitude="39 degree" water_vapor="7.3 um"
        ↪  n_layers="2" zenith_angle="60 degree" resolution="0" temp="240
        ↪  K"/>
    </astroscene>

    <common_optics>
        <optical_component type="Mirror" reflectance="0.965" temp="240 K"
        ↪  comment="M1"/>
        <optical_component type="Mirror" reflectance="0.965" temp="240 K"
        ↪  comment="M2"/>
        <optical_component type="Mirror" reflectance="0.965" temp="240 K"
        ↪  comment="M3"/>
```

```
22      </common_optics>
23
24      <instrument>
25          <optical_component type="Lens" transmittance="0.88" temp="290 K"
            ↪  comment="Dewar Window"/>
26          <optical_component type="Mirror" reflectance="0.97" temp="77 K"
            ↪  comment="Collimator"/>
27          <optical_component type="Mirror" reflectance="0.97" temp="4 K"
            ↪  comment="Dichroic Mirror"/>
28          <optical_component type="Filter"
            ↪  transmittance="data_sofia/filter_FOR_F197.csv" temp="4 K"
            ↪  comment="Filter Wheel FOR_F197"/>
29          <optical_component type="Mirror" reflectance="0.97" temp="4 K"
            ↪  comment="Camera Mirror"/>
30          <optical_component type="Mirror" reflectance="0.97" temp="4 K"
            ↪  comment="Camera Mirror"/>
31          <sensor type="Imager" comment="SWC">
32              <f_number val="5.5"/>
33              <pixel_geometry val="256, 256 pix"/>
34              <center_offset val="0.0, 0.0 pix"/>
35              <pixel>
36                  <quantum_efficiency val="data_sofia/detective_QE.csv"/>
37                  <pixel_size val="50 um"/>
38                  <dark_current val="1.07 electron / (pix * s)"/>
39                  <sigma_read_out val="0.004 electron(1/2) / pix"/>
40                  <well_capacity val="1.6e+7 electron"/>
41              </pixel>
42              <photometric_aperture>
43                  <shape val="circle"/>
44                  <aperture_size val="7 pix"/>
45              </photometric_aperture>
46          </sensor>
47      </instrument>
48  </root>
```

1.: Configuration file used for the simulation of a SOFIA FORECAST observation

F.2. HIFI Configuration File

Listing 2 shows the configuration file used for the simulation of an observation of the HCL^+ towards W31C with Herschel HIFI as described in section 6.2.

```
<root>
    <common>
        <wl_min val="1446 GHz"/>
        <wl_max val="1437 GHz"/>
        <res val="0.23 km/s"/>
        <d_aperture val="3.5 m"/>
        <psf type="Airy" osf="10"/>
        <output path="output" format="fits"/>
        <snr val="42.7"/>
    </common>

    <astroscene>
        <target type="BlackBodyTarget" temp="31.6 K"/>
        <optical_component type="Filter"
        ↪ transmittance="data_w31/hcl_absorption.csv" comment="HCL
        ↪ Absorption"/>
        <optical_component type="CosmicBackground" temp="2.725 K"
        ↪ comment="Cosmic Background"/>
        <optical_component type="CosmicBackground" temp="274 K"
        ↪ comment="Zodical Light"/>
        <optical_component type="CosmicBackground" temp="20 K"
        ↪ comment="Galactic Cirrus"/>
    </astroscene>

    <common_optics>
        <optical_component type="Mirror" reflectance="0.95" temp="88 K"
        ↪ comment="M1"/>
        <optical_component type="Mirror" reflectance="0.95" temp="84 K"
        ↪ comment="M2"/>
    </common_optics>

    <instrument>
        <sensor type="Heterodyne">
            <aperture_efficiency val="0.6"/>
            <main_beam_efficiency val="0.64266"/>
            <receiver_temp val="1050 K"/>
            <eta_fss val="0.96"/>
            <lambda_line val="1444.2 GHz"/>
```

```
32              <lambda_local_oscillator val="1441.5 GHz"/>
33              <kappa val="1"/>
34          </sensor>
35      </instrument>
36  </root>
```

2.: Configuration file used for the simulation of a Herschel HIFI observation

F.3. ESBO Configuration File

Listing 3 shows the configuration file used for the simulation of an observation of the HCL^+ towards W31C with ESBO as described in section 6.2.

```
<root>
    <common>
        <wl_min val="1446 GHz"/>
        <wl_max val="1437 GHz"/>
        <res val="0.23 km/s"/>
        <d_aperture val="5 m"/>
        <psf type="Airy" osf="10"/>
        <output path="output" format="fits"/>
        <snr val="42.7"/>
    </common>

    <astroscene>
        <target type="BlackBodyTarget" temp="31.6 K"/>
        <optical_component type="Filter"
        ↪ transmittance="data_w31/hcl_absorption.csv" comment="HCL
        ↪ Absorption"/>
        <optical_component type="CosmicBackground" temp="2.725 K"
        ↪ comment="Cosmic Background"/>
        <optical_component type="CosmicBackground" temp="274 K"
        ↪ comment="Zodical Light"/>
        <optical_component type="CosmicBackground" temp="20 K"
        ↪ comment="Galactic Cirrus"/>
        <optical_component type="Atmosphere"
        ↪ transmittance="data_w31/transmittance_atmosphere_atran.csv"
        ↪ temp="265 K"/>
    </astroscene>

    <common_optics>
        <optical_component type="Mirror" reflectance="0.95" temp="-15
        ↪ Celsius" obstruction="0.004" obstructor_temp="-20 Celsius"
        ↪ obstructor_emissivity="0.05" comment="M1"/>
        <optical_component type="Mirror" reflectance="0.95" temp="-20
        ↪ Celsius" comment="M2"/>
        <optical_component type="Mirror" reflectance="0.95" temp="-20
        ↪ Celsius" comment="M3"/>
        <optical_component type="Mirror" reflectance="0.95" temp="-20
        ↪ Celsius" comment="M4"/>
```

```
26          <optical_component type="Mirror" reflectance="0.95" temp="-20
            ↪  Celsius" comment="M5"/>
27      </common_optics>
28
29      <instrument>
30          <sensor type="Heterodyne">
31              <aperture_efficiency val="0.55"/>
32              <main_beam_efficiency val="0.67"/>
33              <receiver_temp val="1000 K"/>
34              <eta_fss val="0.97"/>
35              <lambda_line val="1444.2 GHz"/>
36              <lambda_local_oscillator val="1441.5 GHz"/>
37              <kappa val="1"/>
38          </sensor>
39      </instrument>
40  </root>
```

3.: Configuration file used for the simulation of a ESBO simulation

Bibliography

[1] Maier, P. et al. *Towards a European Stratospheric Balloon Observatory: the ESBO design study*. In: *Ground-based and Airborne Telescopes VII*. Ed. by Marshall, H. K.; Spyromilio, J. Vol. 10700. International Society for Optics and Photonics. SPIE, 2018, pp. 1470–1481. DOI: 10.1117/12.2319248.

[2] Gehrz, R. D. et al. *Status of the Stratospheric Observatory for Infrared Astronomy (SOFIA)*. In: *Advances in Space Research* 48.6 (Feb. 5, 2011), pp. 1004–1016. ISSN: 0273-1177. DOI: 10.1016/j.asr.2011.05.023.

[3] Lemke, D. et al. *Balloon-borne telescope THISBE 1: Technology, results, experience*. In: *NASA STI/Recon Technical Report N* 77 (Nov. 1976), p. 30046.

[4] Maier, P. *Partners – ESBO-DS*. University of Stuttgart. 2020. URL: https://esbo-ds.irs.uni-stuttgart.de/wordpress/index.php/partners/ (visited on July 9, 2020).

[5] Klass, L. *Software Requirements Specifications. ESBO Exposure Time Calculator*. Tech. rep. IRS, Mar. 27, 2020.

[6] Burke, B.; Graham-Smith, F. *An Introduction to Radio Astronomy*. 4th ed. Cambridge University Press, 2019. ISBN: 978-1-107-18941-6.

[7] Kazmi, H.; Meixner, M. *SOFIA Observer's Handbook for Cycle 9*. Version 9.1.0. USRA. Apr. 15, 2020. URL: https://www.sofia.usra.edu/science/proposing-and-observing/observers-handbook-cycle-9 (visited on Aug. 29, 2020).

[8] Corle, T. R.; Kino, G. S. *Chapter 1 - Introduction*. In: *Confocal Scanning Optical Microscopy and Related Imaging Systems*. Burlington: Academic Press, 1996, pp. 1–66. ISBN: 978-0-12-408750-7. DOI: https://doi.org/10.1016/B978-012408750-7/50009-4.

[9] Neumann; Bauer; Wolfersdorf. *Wärmestrahlung. Manuskript zur Vorlesung*. 2014.

[10] Hasan, H.; Burrows, C. J. *Telescope Image Modeling (TIM)*. In: *Publications of the Astronomical Society of the Pacific* 107 (Mar. 1995), p. 289. DOI: 10.1086/133552.

[11] Littlefair, S. *Observational Techniques for Astronomers*. University of Sheffield. URL: http://slittlefair.staff.shef.ac.uk/teaching/phy217/lectures/instruments/L14/index.html (visited on Aug. 27, 2020).

[12] Howell, S. B. *Two-Dimensional Aperture Photometry: Signal-to-Noise Ratio of Point-Source Observations and Optimal Data-Extraction Techniques*. In: *Publications of the Astronomical Society of the Pacific* 101 (June 1989), p. 616. DOI: 10.1086/132477.

[13] Klein, U. *Radio astronomy: tools, applications and impacts. Course astro 841*. Argelander-Institut für Astronomie, Bonn. Sept. 2011. URL: https://hera.ph1.uni-koeln.de/ftpspace/simonr/Pablo/Radioastronomy.pdf (visited on Aug. 29, 2020).

[14] Comito, C.; Schilke, P. *Reconstructing reality: Strategies for sideband deconvolution*. In: *Astronomy & Astrophysics* 395.1 (Oct. 2002), pp. 357–371. DOI: 10.1051/0004-6361:20021277.

[15] Teyssier, D. et al. *The Heterodyne Instrument for the Far Infrared (HIFI) Handbook*. Version 1.2. ESA. Herschel Science Centre, Dec. 18, 2017.

[16] Johannes Ernesti, P. K. *Python 3*. Rheinwerk Verlag GmbH, Sept. 1, 2020. ISBN: 3836279266.

[17] Meyer, B. *Object-oriented software construction*. Upper Saddle River, N.J: Prentice Hall PTR, 1997. ISBN: 9780136291558.

[18] Gamma, E. et al. *Design Patterns. Elements of Reusable Object-Oriented Software*. MITP Verlags GmbH, Jan. 26, 2015. 480 pp. ISBN: 3826697006.

[19] Klass, L. *Software Verification. ESBO Exposure Time Calculator*. Tech. rep. IRS, July 23, 2020.

[20] Uslenghi, M.; Falomo, R.; Fantinel, D. *AETC: a powerful web tool to simulate astronomical images*. In: *Modeling, Systems Engineering, and Project Management for Astronomy VI*. Ed. by Angeli, G. Z.; Dierickx, P. SPIE, Aug. 2016. DOI: 10.1117/12.2233621.

[21] Krabbe, A. *SOFIA telescope*. In: *Airborne Telescope Systems*. Ed. by Melugin, R. K.; Roeser, H.-P. SPIE, June 2000. DOI: 10.1117/12.389103.

[22] Herter, T. L. et al. *FORCAST: A Mid-Infrared Camera for SOFIA*. In: *Journal of Astronomical Instrumentation* 07.04 (Dec. 2018), p. 1840005. DOI: 10.1142/s2251171718400056.

[23] White, T. R. et al. *Beyond the Kepler/K2 bright limit: variability in the seven brightest members of the Pleiades*. In: *Monthly Notices of the Royal Astronomical Society, 471, p.2882-2901 (2017)* (Aug. 24, 2017). DOI: 10.1093/mnras/stx1050.

[24] Fixsen, D. J. *The Temperature of the Cosmic Microwave Background*. In: *The Astrophysical Journal* 707.2 (Nov. 10, 2009), pp. 916–920. DOI: 10.1088/0004-637X/707/2/916.

[25] Adams, J. D. et al. *FORCAST: a first light facility instrument for SOFIA*. In: *Ground-based and Airborne Instrumentation for Astronomy III*. Ed. by McLean, I. S.; Ramsay, S. K.; Takami, H. SPIE, July 2010. DOI: 10.1117/12.857049.

[26] Klass, L.; Maier, P. *Code for the Simulation of SOFIA/FORECAST in "ESBO-ETC: The modular open-source Exposure Time Calculator"*. Version v1. Oct. 2020. DOI: 10.24433/CO.9752252.v1.

[27] Gerin, M. et al. *Probing InterStellar Molecules with Absorption line Studies (PRISMAS)*. In: Mar. 25, 2008.

[28] Luca, M. D. et al. *Herschel / HIFI Discovery of HCl^+ in the Interstellar Medium*. In: *The Astrophysical Journal* 751.2 (May 2012), p. L37. DOI: 10.1088/2041-8205/751/2/l37.

[29] Teyssier, D. *Quick-Start Guide to Herschel-HIFI*. Version 2.1. ESA. Herschel Science Centre, Jan. 16, 2018.

[30] Exter, K. et al. *The Photodetector Array Camera and Spectrometer (PACS) Handbook*. Version 4.0.1. ESA. Herschel Science Centre, Apr. 30, 2019.

[31] Klass, L.; Maier, P. *Code for the Simulation of Herschel/HIFI in "ESBO-ETC: The modular open-source Exposure Time Calculator"*. Version v1. Oct. 2020. DOI: 10.24433/CO.3100674.v1.

[32] Reach, W. T. et al. *The mid-infrared spectrum of the zodiacal and exozodiacal light*. In: *Icarus* 164.2 (Aug. 2003), pp. 384–403. DOI: 10.1016/s0019-1035(03)00133-7.

[33] Leinert, C. et al. *The 1997 reference of diffuse night sky brightness*. In: *Astronomy and Astrophysics Supplement Series* 127.1 (Jan. 1998), pp. 1–99. DOI: 10.1051/aas:1998105.

[34] Sein, E. et al. *A 3.5m diameter Sic telescope for Herschel mission*. In: *IR Space Telescopes and Instruments*. Ed. by Mather, J. C. Vol. 4850. International Society for Optics and Photonics. SPIE, 2003, pp. 606–618. DOI: 10.1117/12.461804.

[35] *Standard Atmosphere*. Tech. rep. ISO 2533:1975. ISO - International Organization for Standardization, May 15, 1975. 108 pp.

[36] Maier, P. *STUDIO Thermal Control System*. Tech. rep. Version 1.00. University of Stuttgart, Oct. 17, 2019.

[37] Klass, L.; Maier, P. *Code for the Simulation of ESBO/GREAT in "ESBO-ETC: The modular open-source Exposure Time Calculator"*. Version v1. Oct. 2020. DOI: 10.24433/CO.9321102.v1.

[38] Zombeck, M. V. *Handbook of Space Astronomy and Astrophysics*. 3rd ed. Cambridge University Press, 2006. DOI: 10.1017/CBO9780511536359.