

Raumfahrttechnik und Extraterrestrik

Advances in
Space Technology
and Exploration

Julius-Maximilians-

**UNIVERSITÄT
WÜRZBURG**

Nr. 1

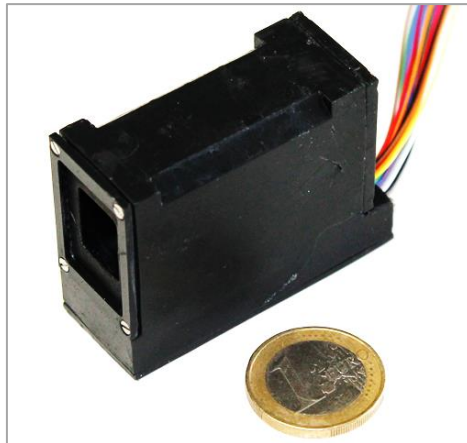
Oleksii Balagurin

Designoptimierung von
Sternsensoren für Pico-
und Nanosatelliten

Designoptimierung von Sternsensoren für Pico- und Nanosatelliten

Dissertation zur Erlangung des naturwissenschaftlichen Doktorgrades der Julius-Maximilians-Universität Würzburg

vorgelegt von
Oleksii Balagurin
aus Würzburg



Würzburg, März 2022

Eingereicht am:

Bei der Fakultät für Mathematik und Informatik

Vorsitzender: Prof. Dr. Reiner Kolla

1. Gutachter: Prof. Dr.-Ing. Hakan Kayal

2. Gutachter: Prof. Dr.-Ing. Klaus Briß

Tag der mündlichen Prüfung: 03.12.2021

Danksagung

Als erstes möchte ich mich bei meinem Erstbetreuer Herrn Professor Hakan Kayal für die jahrelange, erfahrungsreiche und produktive Zusammenarbeit im Bereich Raumfahrt bedanken.

Weiterhin möchte ich mich bei Professor Klaus Briß für die Unterstützung und die Übernahme der Rolle des Gutachters bedanken.

Extradank gilt auch meinen Kollegen der Professur für Raumfahrttechnik, die mir während der Entstehung dieser Arbeit mit Rat und Tat zur Seite gestanden haben. Besonders hervorzuheben ist die sprachliche und fachbezogene Korrekturarbeit von Herrn Tobias Schwarz.

Außerdem bedanke ich mich bei allen Studierenden für wertvolle Beiträge während ihrer Tätigkeit als Hilfwissenschaftler oder in Form von Abschlussarbeiten.

Für ihre Geduld und weitgehende persönliche Unterstützung danke ich meiner Ehefrau Tatjana und meinen beiden Töchtern Clara und Sophie.

Kurzfassung

Die Raumfahrt ist eine der konservativsten Industriebranchen. Neue Entwicklungen von Komponenten und Systemen beruhen auf existierenden Standards und eigene Erfahrungen der Entwickler. Die Systeme sollen in einem vorgegebenen engen Zeitrahmen projektiert, in sehr kleiner Stückzahl gefertigt und schließlich aufwendig qualifiziert werden. Erfahrungsgemäß reicht die Zeit für Entwicklungsiterationen und weitgehende Perfektionierung des Systems oft nicht aus. Fertige Sensoren, Subsysteme und Systeme sind Unikate, die nur für eine bestimmte Funktion und in manchen Fällen sogar nur für bestimmte Missionen konzipiert sind. Eine Neuentwicklung solcher Komponenten ist extrem teuer und risikobehaftet. Deswegen werden flugerprobte Systeme ohne Änderungen und Optimierung mehrere Jahre eingesetzt, ohne Technologiefortschritte zu berücksichtigen.

Aufgrund des enormen finanziellen Aufwandes und der Trägheit ist die konventionelle Vorgehensweise in der Entwicklung nicht direkt auf Kleinsatelliten übertragbar. Eine dynamische Entwicklung im Low Cost Bereich benötigt eine universale und für unterschiedliche Anwendungsbereiche leicht modifizierbare Strategie. Diese Strategie soll nicht nur flexibel sein, sondern auch zu einer möglichst optimalen und effizienten Hardwarelösung führen.

Diese Arbeit stellt ein Software-Tool für eine zeit- und kosteneffiziente Entwicklung von Sternsensoren für Kleinsatelliten vor. Um eine maximale Leistung des Komplettsystems zu erreichen, soll der Sensor die Anforderungen und Randbedingungen vorgegebener Anwendungen erfüllen und darüber hinaus für diese Anwendungen optimiert sein. Wegen der komplexen Zusammenhänge zwischen den Parametern optischer Sensorsysteme ist keine „straightforward“ Lösung des Problems möglich. Nur durch den Einsatz computerbasierter Optimierungsverfahren kann schnell und effizient ein bestmögliches Systemkonzept für die gegebenen Randbedingungen ausgearbeitet werden.

„Die meisten Probleme entstehen bei Ihrer Lösung.“

Leonardo da Vinci

Inhaltsverzeichnis

1	Abkürzungsverzeichnis	8
2	Einleitung.....	11
3	Sternsensoren als System	19
3.1	<i>Aufbau und Funktionsprinzip eines Sternsensors.....</i>	<i>19</i>
3.2	<i>Motivation und Problemformulierung.....</i>	<i>20</i>
3.2.1	Multiparameter- und multikriterielles Problem in der Entwicklung.....	20
3.2.2	Multidisziplinarität des Designproblems	21
3.2.3	Pico- und Nanosatelliten-Aspekte in der Entwicklung von Sternsensoren.....	21
4	Besonderheiten der Entwicklungsprozesse in der Raumfahrt	23
4.1	<i>Entwicklung nach Projektphasen</i>	<i>23</i>
4.2	<i>Besonderheiten bei Pico- und Nanosatellitenprojekten</i>	<i>24</i>
4.3	<i>Top-Down-Verfahren.....</i>	<i>26</i>
4.4	<i>Iterative Entwicklung.....</i>	<i>27</i>
4.5	<i>STELLA Sternsensor als Beispiel der iterativen Entwicklung.....</i>	<i>27</i>
4.6	<i>Ziele der Arbeit.....</i>	<i>32</i>
5	Modellbasierter Entwurf eines optimalen Sternsensordesigns	33
5.1	<i>Modellbasierte und optimierende Entwicklungsstrategie</i>	<i>33</i>
5.2	<i>Konzept des automatisierten Systementwurfs.....</i>	<i>34</i>
5.3	<i>AROS – Forschungsvorhaben zur Verbesserung des Designablaufs von Sternsensoren.....</i>	<i>35</i>
5.4	<i>Struktur des Optimierungstools</i>	<i>36</i>
5.5	<i>Modell des Sternsensordesigns</i>	<i>36</i>
5.5.1	Abbildung der Sterne	37
5.5.2	Bildsensor.....	42
5.5.3	Paraxiale Optik.....	50
5.5.4	Nichtparaxiale Optik und optische Aberrationen.....	58
5.5.5	Auflösung des gesamten opto-elektronischen Systems	69
5.5.6	Verteilung der Sterne am Himmel	71
5.5.7	Bild- und Datenverarbeitung.....	76
5.6	<i>Systemoptimierung.....</i>	<i>99</i>
5.6.1	Problembeschreibung bei der Multiparameter-Optimierung	100
5.6.2	Klassifikation der Optimierungsmethoden.....	101
5.6.3	Auswahl des Optimierungsverfahrens zum Sternsensordesign	103
5.6.4	Evolutionstrategie	104
5.6.5	Multikriterielle Optimierungsmethoden.....	109
6	Implementierung des Optimierungstools für die automatische Entwicklung von Sternsensoren.....	115
6.1	<i>Implementierung des Tools.....</i>	<i>115</i>
6.2	<i>Parametrierung und Initialisierung des Tools</i>	<i>116</i>
6.3	<i>Verifikation der Software.....</i>	<i>117</i>

6.4	<i>Parallelverarbeitung</i>	122
7	Anwendung des Tools bei der Entwicklung von Sternsensoren für Pico- und Nanosatelliten	124
7.1	<i>Ziele der Entwicklung und der Optimierung</i>	124
7.1.1	Optimierung 1 auf die Bedürfnisse von Picosatelliten (OP1)	125
7.1.2	Optimierung 2 auf die Bedürfnisse von Nanosatelliten (OP2)	125
7.2	<i>Anforderungen und Randbedingungen für OP1 und OP2</i>	126
7.3	<i>Elektrische Hardware</i>	127
7.3.1	Bildsensoren	127
7.3.2	Systemarchitektur OP1	129
7.3.3	Systemarchitektur OP2.....	130
7.4	<i>Ergebnisse der Optimierung</i>	131
7.5	<i>Bau und Test der Prototypen</i>	134
7.5.1	Optik.....	134
7.5.2	Mechanische Struktur.....	135
7.5.3	Elektronik	136
7.6	<i>Funktionale Tests der Prototypen der Sternsensoren</i>	137
7.7	<i>Verbesserung der Sternidentifikation der AROS-Sternsensoren</i>	138
7.8	<i>Qualifikationsprogramm</i>	139
7.9	<i>Technische Spezifikation der OP1- und OP2-Sternsensoren</i>	140
8	In-Orbit-Verifikation auf SONATE	142
8.1	<i>SONATE-Satellit</i>	142
8.1.1	Nutzlasten.....	142
8.1.2	Satellitenbus	143
8.2	<i>Bodensegment und Satellitenbetrieb</i>	148
8.3	<i>Experimente mit den OP1 Sternsensoren</i>	150
9	Zusammenfassung und Ausblick	154
10	Literaturverzeichnis	157
11	Anhang	167

1 Abkürzungsverzeichnis

ADCS	Attitude Determination and Control System
AHB	Advanced High-performance Bus
APS	Active Pixel Sensors
AR	Acceptance Review
BMWi	Bundesministerium für Wirtschaft und Energie
BNA	Bundes Netz Agentur
BSI	Back Side Illumination
CAD	Computer-Aided Design
CAE	Computer Aided Engineering
CAN	Controller Area Network
CCD	Charge-Coupled Devices
CCTV	Closed Circuit Television
CDR	Critical Design Review
CMOS	Complementary Metal-Oxide Semiconductor
COTS	Commercial Off-The-Shelf
CPU	Central Processing unit
CRR	Commissioning Results Review
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DSP	Digital Signal Processor
ECSS	European Cooperation for Space Standardization
EGSE	Electrical Ground Support Equipment
ESA	European Space Agency
FKZ	Förderkennzeichen
FOV	Field Of View
FPGA	Field Programmable Gate Arrays
FRR	Flight Readiness Review

GUI	Graphical User Interface
ICD	Interface Control Document
ICRF	International Celestial Reference Frame
ITAR	International Traffic in Arms Regulations
ITU	International Telecommunication Union
LEO	Low Earth Orbit
LEOP	Launch and Early Orbit Phase
LIS	Lost In Space
MCR	Mission Close-out Review
MDR	Mission Definition Review
MEMS	Microelectromechanical Systems
MOGA	Multi Objektive Genetic Algorithm
MPPT	Maximum Power Point Tracking
MRAM	Magnetoresistive Random Access Memory
NASA	National Aeronautics and Space Administration
NFOV	Narrow Field of View
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
OBC	Boardcomputer
OBDH	On Board Data Handling
OP	Optimierung
ORR	Operational Readiness Review
PDH	Payload Data Handling
PDR	Preliminary Design Review
PRR	Preliminary Requirements Review
PSF	Point Spread Function
QR	Qualification Review

QUEST	Quaternion Estimator
RAM	Random-Access Memory
RMS	Root Mean Square
SDR	Software Defined Radios
SPEA	Strength Pareto Evolutionary Algorithm
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRR	System Requirements Review
SSTV	Slow Scan Television
TC	Telecommando
TID	Total Ionizing Dose
TJT	Triple-Junction-Technologie
TM	Telemetry
TQFP	Thin Quad Flat Pack
TV	Thermal-Vakuum
UHF	Ultra High frequency
VEGA	Vector Evaluted Genetic Algorithm
VHF	Very High Frequency
WFOV	Wide Field of View

2 Einleitung

Beim Navigieren durch die Straßen mit unserem Smartphone, beim Abrufen der aktuellen Wettervorhersage oder beim Anschauen einer digitalen und hochauflösenden Fernsehsendung denken wir nur selten darüber nach, was im Hintergrund dieser Technik steht und wie aufwändig und kostspielig sie ist. Dennoch können wir uns unser alltägliches Leben ohne diesen Komfort nicht mehr vorstellen. Hoch über der Erdoberfläche, verborgen vor dem menschlichen Auge, fliegen Satelliten, die diese unverzichtbaren Dienste bereitstellen oder die dafür notwendigen Informationen liefern. Weil Satelliten den extremen Bedingungen des Weltalls wartungslos standhalten müssen, ist die verwendete Hardware hochtechnologisch, umfangreich und extrem teuer. Aber bevor ein Satellit seinen Dienst im Orbit aufnehmen kann, muss er zuerst mit einer Rakete ins All befördert werden. Je nach Satellitenmission nehmen die Startkosten einen signifikanten Anteil des Budgets ein und sind proportional zur Satellitenmasse. Aus diesem Grund ist die Miniaturisierung eine treibende Kraft in der Entwicklung von Raumfahrtssystemen.

Mit der Entwicklung von neuen kompakten Halbleiterkomponenten entstanden Ende der 90er Jahre zwei neue Satellitenklassen: Picosatelliten mit einer Masse von bis zu 1 kg und Nanosatelliten - bis etwa 20 kg (Abbildung 1). Diese Satelliten können kostengünstig als „piggyback“-Passagiere zusammen mit einem größeren Hauptsatellit gestartet werden. Auch die Entwicklungszeit von solchen Kleinsatelliten ist sehr attraktiv und beträgt im Durchschnitt zwei Jahre [RIC15].

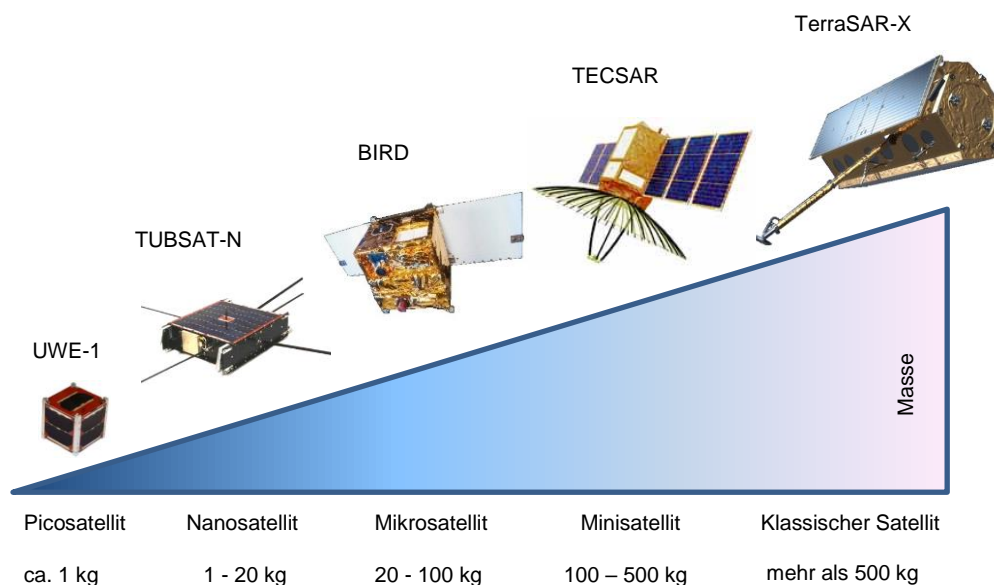


Abbildung 1. Satellitenklassen [BAL18]

Eine besondere Bauform der Pico- und Nanosatelliten sind CubeSats. Diese Satelliten werden nach einheitlichen Standards gebaut und in speziellen Auswurfscontainern gestartet. Die Größe eines CubeSats wird in Units gemessen. Eine Unit (1U) entspricht einem Würfel mit einer Kantenlänge von 10cm und einer Masse von 1 kg. Der Ursprung dieser Bauform liegt bereits im Jahre 1999. Der CubeSat-Standard wurde von zwei universitären Gruppen aus der Stanford University und der California Polytechnic State University als Ausbildungsplattform zum

Erlernen des Entwicklungsprozesses, der Handhabung, der Startprozedur und des Betriebs von Raumfahrtssystemen eingeführt [WET2020]. Seitdem wurden mehr als 1000 CubeSats ins Weltall befördert. Und die Prognosen zeigen weitere drastisch steigende Tendenzen in der Anzahl der Nanosatellitenstarts [VIL19]. Im Vergleich zu 2017-2018 wird sich die Anzahl der Starts in den nächsten zwei Jahren verdoppeln (Abbildung 2).

Die führenden Raumfahrtunternehmen im Kleinsatellitenbereich, wie Planet Labs und Spire, haben die Vorteile der CubeSats frühzeitig erkannt und setzen diese in Konstellationen von hunderten Satelliten operational ein. Seit 2014 überwiegt der kommerzielle Anteil an CubeSat-Missionen [SWA19]. Auch die Raumfahrtagenturen wie DLR, NASA, ESA und Roskosmos entwickeln eigene Kleinsatelliten oder unterstützen vielseitig die Entwicklungen an den Universitäten und in der Industrie.

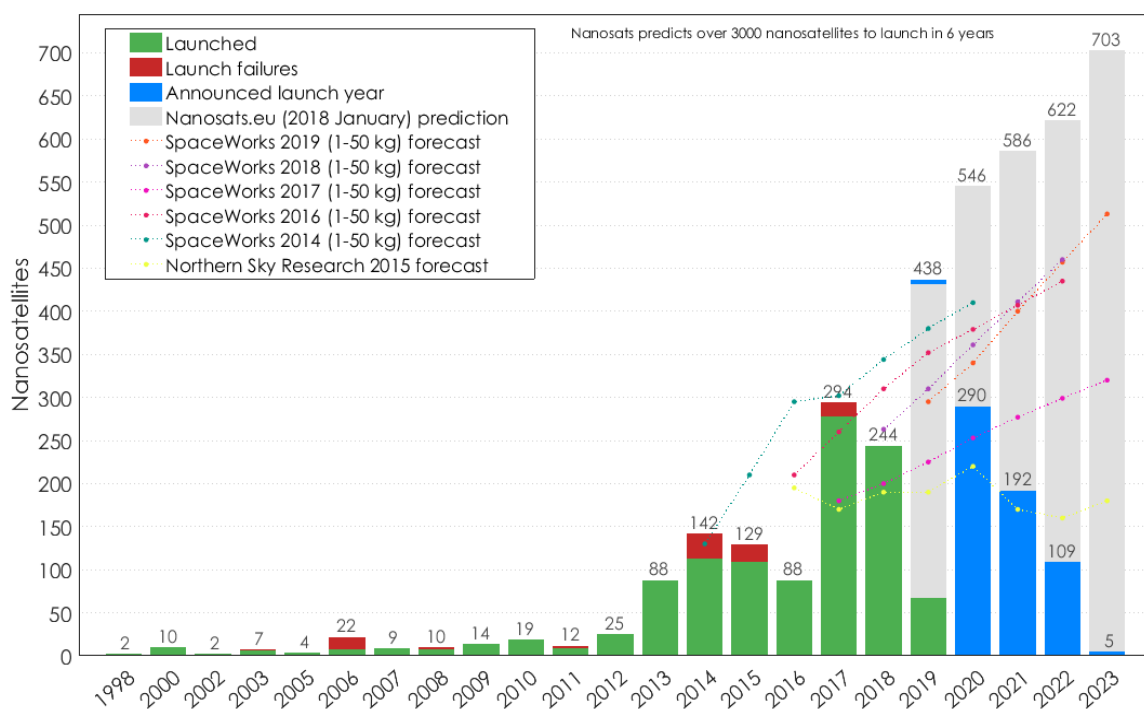


Abbildung 2. Anzahl der Nanosatellitenstarts 2019 mit Prognose [NAN19]

Im Rahmen der derzeit massiv fortschreitenden Entwicklung der „new space economy“ werden mittelfristig viel mehr miniaturisierte und vor allem agile Satellitenkomponenten und -systeme benötigt. Der Bedarf an solchen Systemen für kommerzielle Weltraumanwendungen wird massiv ansteigen.

Während die ersten Pico- und Nanosatelliten ausschließlich für Ausbildungszwecke und als Technologiedemonstratoren gebaut und betrieben wurden, entwickeln sich diese Satelliten in den Jahren zu Plattformen für ernste wissenschaftliche Experimente in diversen Bereichen, wie z.B. Astronomie, Erdbeobachtung, Erforschung der Atmosphäre und Kommunikation, weiter [NAT16]. Dabei müssen die Satellitenkomponenten, der sogenannte Satellitenbus, immer härtere Anforderungen erfüllen, um die wissenschaftlichen Instrumente betreiben zu können. Vom Entwicklungsniveau und der Komplexität des Gesamtsystems nähern sich die Pico- und Nanosatelliten ihren großen Brüdern an. Bei der Entwicklung von neuen Miniatur-Systemen für Kleinsatelliten soll die in großen Satelliten verwendete Technik in kurzer Zeit

für die Anwendung in Kleinsatelliten adaptiert werden. Dabei herrschen etwas andere Randbedingungen im Vergleich zu den großen, tonnenschweren Satelliten: Volumen und Gewicht, Energieverbrauch und Entwicklungskosten sowie Entwicklungszeit sind weitgehend zu minimieren.

Aufbau des Satellitenbusses

Moderne Pico- und Nanosatelliten haben alle notwendigen Subsysteme, die auch in großen Satelliten vorhanden sind. Die Konfiguration und Komplexität des jeweiligen Subsystems sind abhängig von der Nutzlast bzw. dem Experiment an Bord des Satelliten. Die zwei nachfolgenden Beispiele erfolgreicher Nanosatellitenmissionen mit einem wissenschaftlichen Hintergrund zeigen den möglichen Aufbau des Satellitenbusses:

Die **BRITE** (BRiGht-star Target Explorer) Konstellation ist ein Netzwerk aus fünf Satelliten, die 2013 und 2014 in einen niedrigen Erdorbit zwischen 600 und 800 km Höhe mit einer Inklination von 97-100° gebracht wurden. Die Hauptaufgabe der Konstellation ist die photometrische Messung der Schwankungen von Sternhelligkeiten. Als Instrument wird ein kleines Teleskop mit einer Öffnung von 30mm und einem Sichtfeld von 24° verwendet [KAI08]. Die Beobachtungen werden in zwei Spektralbereichen, blau (390-460nm) und rot (550-700nm), durchgeführt. Der eingesetzte Interline CCD-Detektor von Kodak hat eine Auflösung von 11 Mpixel. Es sollen 286 Sterne mit einer Magnitude heller als 3,5 aufgenommen werden. Die Dauer einer Aufnahme beträgt 15 min. pro Orbit, mit einer Belichtungszeit von ca. 1 s.

Die BRITE-Satelliten bestehen aus allen herkömmlichen Subsystemen [WEI13, KIE04]:

1. Die Struktur eines BRITE-Satelliten ist quaderförmig, mit einer Seitenlänge von 20 cm und ist aus Aluminium gefertigt. Wie bei einem 1U-CubeSat befinden sich die Kontaktflächen an vier Schienen, die entlang der Quaderseiten angebracht sind (Abbildung 3). Die Gesamtmasse des Satelliten beträgt 7 kg.

2. Der Bordcomputer (OBC) basiert auf einem ARM7-Prozessor, der mit 40 MHz getaktet ist. Der OBC hat 256 KB Flash-Speicher für Programme, 1 MB SRAM für die Verarbeitung der Bilddaten und Erfassung der Telemetrie sowie 256 MB Flash für die Aufzeichnungen von offline Daten. Insgesamt sind drei identische Leiterplatten verbaut, die eine von drei Funktionalitäten übernehmen: OBC, Lageregelungskontroller oder Payload Data Handling (PDH).

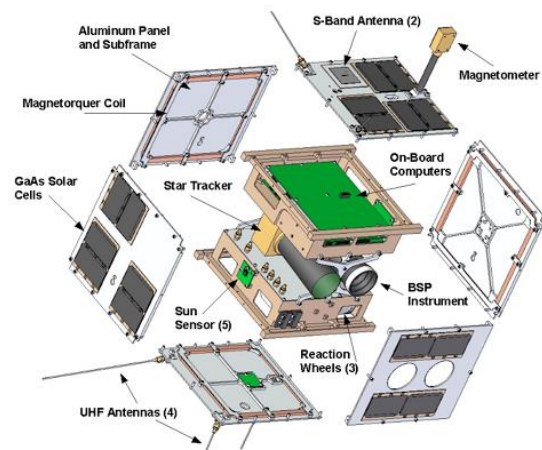
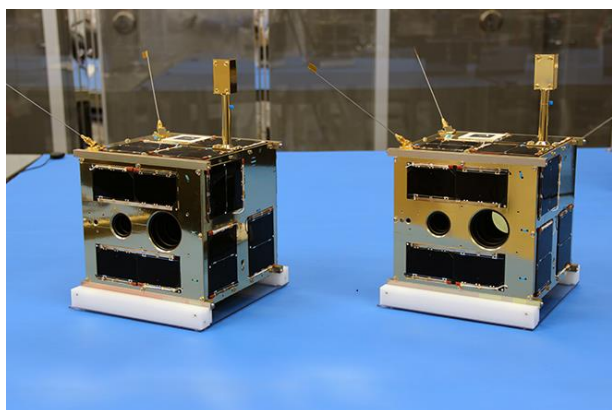


Abbildung 3. BRITE Satelliten und deren Bus [PAB16, SCH10]

3. Die Energieversorgung aller elektronischen Komponenten erfolgt über einen unregelmäßig +4V Bus. Als Energiequelle dienen Mehrschicht-Solarzellen mit Triple-Junction-Technologie (TJT). Fast jede Seite des Satelliten hat sechs Solarzellen, die bis zu 5W elektrische Leistung generieren. Für die Speicherung der elektrischen Energie ist eine Li-Ion Batterie mit einer Kapazität von 4,8 Ah zuständig.

4. Für die Kommunikation mit den BRITE-Satelliten werden zwei Frequenzbänder verwendet. Die Kommandierung erfolgt über vier Monopolantennen und einem UHF-Empfänger. Die Uplinkrate beträgt 9.6 kbit/s. Der Downlink erfolgt über zwei festmontierte Patch-Antennen und einen nachgeschalteten S-Band Transmitter mit max. 256 kbit/s.

5. Die Thermalkontrolle des BRITE-Satelliten ist passiv. Durch eine entsprechende Beschichtung der Außenseiten wird die Temperatur des Satelliten im Bereich von +10°C bis +40°C gehalten.

6. Die BRITE-Satelliten sind drei-Achsen-stabilisiert. Dies bedeutet, dass die Orientierung um die drei Achsen des Satelliten unabhängig voneinander geregelt werden kann. Um die gestellten wissenschaftlichen Anforderungen der Mission zu erfüllen, wurde ein spezielles Lageregelungssystem (Attitude Determination and Control System, ADCS) von der SFL, Canada konzipiert. Das ADCS erreicht eine Ausrichtungsgenauigkeit von 1,5" rms. Die grobe Lagebestimmung wird anhand von sechs Sonnensensoren und einem Dreiachsen-Magnetometer ermittelt. Für die Feinausrichtung wurde ein Sternsensor parallel zum Teleskop eingebaut. Als Aktuatoren werden sechs Luftspulen, welche in die Solarpanels integriert sind, und drei Reaktionsräder verwendet. Die Lageregelungssoftware läuft auf einem zum OBC baugleichen Bord.

Die **AeroCubes-7 (OCSD-B und -C)** sind zwei 1,5U-CubeSats, die eine Technologiedemonstration im Bereich der optischen Kommunikation durchführen. Die Satelliten wurden im Dezember 2017 in einen 450 km Orbit gebracht. Die Hauptnutzlast der beiden Satelliten ist ein Laser-Transceiver mit einer Sendeleistung von bis zu 2 W bei einer Wellenlänge von 1064 nm. Der Laser-Transceiver erlaubt die Kommunikation mit der Bodenstation mit einer Datenrate von bis zu 100 Mbit/s. Außerdem haben die Satelliten drei Farbkameras mit unterschiedlichen Sichtfeldern und Orientierungen.

Um die Kommunikationsexperimente durchführen zu können, wurde ein sehr anspruchsvoller Satellitenbus entwickelt [JAN13, JAN15, JAN16, ROW18]:

1. Die Struktur des Satelliten wurde nach dem CubeSat-Standard mit einer Größe von 1,5U gebaut. Die Satelliten haben zwei Solarpaneele, die nach dem Auswurf des Satelliten entfaltet wurden (Abbildung 4). Die Gesamtmasse eines AeroCubes beträgt 2,3 kg.

2. Die Satelliten nutzen ein verteiltes Rechnersystem, bestehend aus über 20 Mikroprozessoren und drei Field Programmable Gate Arrays (FPGA). Diese übernehmen die Rolle des OBCs und weitere Aufgaben wie an-Bord-Berechnungen für Sensorik, Nutzlast, Lagerfassung und -Regelung. Außerdem stehen 8 GByte Flash-Speicher zur Datenspeicherung zur Verfügung.

3. Um den Laser-Transmitter betreiben zu können, wurde ein Hochleistungsenergieversorgungssystem entwickelt, das in der Lage ist, bis zu 60 W über einen Zeitraum von mehr als 3 min. zu liefern. Die TJT-Solarzellen produzieren durchschnittlich 4,5 W elektrische Leistung. Die gewonnene Energie wird für die Schattenzeiten in zwei kommerziellen 18650 Li-Ion-Zellen mit einer Kapazität von 14Wh gespeichert. Zwei weitere Hochstrom-Akku-Zellen wurden ausschließlich für die Versorgung des Lasers verbaut.

4. Für die Kommandierung und zum Aussenden der Telemetrie stehen zwei Transceiver im 915 MHz-Band mit einer einstellbaren Datenrate von bis zu 0,5 Mbit/s, zur Verfügung. Einer der beiden Transceiver ist auf der Basis eines neu entwickelten Software Defined Radios (SDR) realisiert.

5. Wie auch bei den BRITE-Satelliten ist die Thermalkontrolle der AeroCubes passiv. Es wurden keine besonderen Maßnahmen für zusätzliche Aufnahme bzw. Abfuhr von Wärme getroffen.

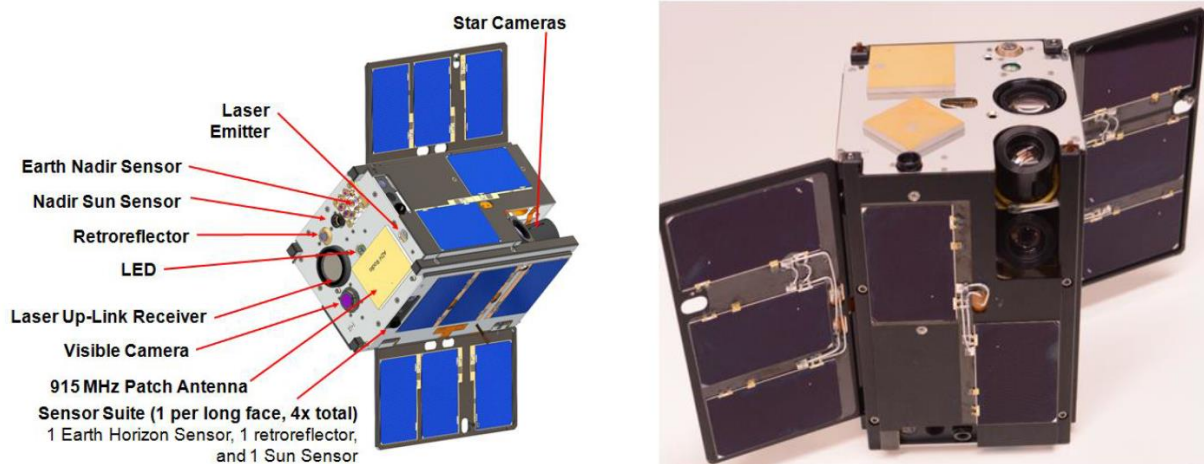


Abbildung 4. Satellitenbus von AeroCube-7 [JAN16]

6. Die beide AeroCubes besitzen ein sehr ausgeprägtes System zur Lage- und Positionserfassung sowie deren Kontrolle. Die exakte Position der Satelliten wird durch die eingebauten GPS-Empfänger ermittelt. Außerdem können die angebrachten Retroreflektoren für die Bestimmung der Position verwendet werden. Um den Orbit bzw. das Auseinandertriften der Satelliten aktiv kontrollieren zu können, ist ein Wasserdampftrieb eingebaut. Die Lage wird mit mehreren Sensoren erfasst: sechs zweiachsige Sonnensensoren, zwei dreiachsige Magnetometer, ein Erdhorizontsensor und zwei Sternsensoren sind dafür vorgesehen. Die Drehraten werden mit zwei dreiachsigen MEM-Gyros erfasst. Für die aktive Drehung der Satelliten sorgen drei Reaktionsräder und drei Magnetspulen. Die erreichbare Ausrichtungsgenauigkeit liegt bei $0,15^\circ$, dies entspricht den Anforderungen des Laser-Kommunikationsexperiments.

Diese beiden Beispiele zeigen nur einen winzig kleinen Auszug aus der unglaublichen Menge von erfolgreichen Pico- und Nanosatellitenmissionen und sollen illustrieren, wie solche Satelliten kleine, aber beliebig komplizierte Experimente kostengünstig durchführen können. Während der Experimente ist die Ausrichtungsgenauigkeit der Instrumente bzw. der Nutzlast einer der wichtigsten Aspekte der Mission. Daher wird hier bei der Implementierung des Lagerege-

lungssystems besonders viel Aufwand betrieben, um die initialen Anforderungen an die Satellitenlage unter Berücksichtigung der Randbedingungen der Pico- und Nanosatelliten erfüllen zu können.

Lageregelungssystem (ADCS)

Die primäre Aufgabe eines aktiven ADCS ist es den Satelliten in die Richtung des vorgegebenen Zielobjekts mit gewünschter Genauigkeit und Geschwindigkeit zu orientieren. Je nach Mission werden beim ADCS verschiedene Betriebsmodi implementiert. Ein Teil der Modi deckt Bedürfnisse des Routinebetriebs und dient der Lebenserhaltung des Satelliten, z.B. „Sun Pointing“ - Ausrichtung der Solarpaneele in die Richtung der Sonne, um die Batterie aufladen zu können. Andere beziehen sich auf Experimente mit einer Nutzlast. In diesen Modi, wie z.B. „Accurate Target Pointing“, sind die Anforderungen an die Genauigkeit und Jitter besonders strikt.

Üblicherweise besteht ein ADCS aus drei Hauptbestandteilen: aus einem Lageregler, aus einem oder mehreren Aktuatoren sowie aus einem Lageerfassungssystem mit diversen Sensoren.

Der Lageregler ist das Gehirn des ADCS. Er analysiert die Abweichung des gemessenen und des kommandierten Werts und steuert anhand bestimmter Regelstrategien die Aktuatoren. Die Software-Implementierung des Lagereglers kann entweder ein Teil der OBC-Software sein oder auf eigener spezifischer Hardware laufen.

Aktuatoren wandeln die Signale des Lagereglers, sogenannte Stellgrößen, in die gewünschte Drehbewegung um eine bestimmte Achse um. Zu den typischen Aktuatoren gehören Magnetspulen, Reaktionsräder bzw. gefesselte Kreisel und ADCS-Antriebe. Magnetspulen erzeugen proportional zur Stromstärke ein Magnetfeld, das mit dem natürlichen Erdmagnetfeld interagiert und eine Drehung des Satelliten hervorruft. Alleine mit den Spulen können nur zwei Achsen eines Satelliten dynamisch geregelt werden. Um ein dreiaachsenstabilisiertes System zu erreichen, werden die Magnetspulen in Kombination mit Reaktionsrädern verwendet. Reaktionsräder sind elektrische Motoren mit einer Schwungmasse. Beim Beschleunigen der Schwungmasse wird der Drehimpuls auf den Satelliten übertragen und setzt ihn in Drehbewegung. Eine andere aber in Pico- und Nanosatelliten selten eingesetzte Alternative zu Reaktionsrädern sind Flüssigkeitsdrallringe oder elektrische bzw. chemische Miniaturantriebe.

Lageregelungssensoren erfassen die Orientierung des Satelliten im Weltraum. Das Messprinzip dieser Sensoren basiert auf der Ermittlung der Richtung zu bekannten Referenzobjekten im Weltall, beispielweise von der Sonne, Sterne oder der Erde selbst. Auch physikalische Größen, wie das Erdmagnetfeld, die gut erforscht und kartographiert sind, können als Referenzobjekte für die Lageberechnung verwendet werden. Um die Lage eindeutig zu bestimmen, benötigt man mindestens zwei Messvektoren, also die Richtung von zwei bekannten Objekten im eigenen körperfesten sowie im Referenzkoordinatensystem. Die in Pico- und Nanosatelliten häufig eingesetzten Lageregelungssensoren, welche bereits in den zwei vorangegangenen Beispielmmissionen erwähnt wurden, sind: Sonnensensoren, Magnetfeldsensoren sowie Stern- und Horizontsensoren. Da diese Sensoren verschiedene Messprinzipien nutzen,

sind sie unterschiedlich genau und haben gewisse Vor- und Nachteile (Tabelle 1). Um die gewünschte Leistung in der Lageerfassung und Datenkontinuität zu erreichen, werden verschiedene Sensortypen zu einem Messsystem kombiniert. Außerdem werden mehrere Sensoren des gleichen Typs verbaut, um die Sichtbarkeit des Referenzobjektes zu maximieren. So muss mindestens ein Sonnensensor auf jeder Seite eines quaderförmigen Satelliten angebracht werden, um die Sonne immer im Blick zu behalten, auch wenn der Satellit sich dreht. In größeren Satelliten sind auch mehrere Sternsensoren üblich, weil abhängig von der aktuellen Satellitenlage ein Teil des Sternenhimmels entweder durch die Erde verdeckt oder der Detektor durch die Sonne geblendet ist. Hinzu kommt ein Redundanzkonzept, das unter Umständen die Anzahl der Sensoren verdoppelt. Gyroskope sind keine direkten Lageerfassungssensoren. Sie messen die Drehraten des Satelliten z. B. für eine Drehratenregelung oder helfen bei der Prädiktion der Lage, wenn der eine oder andere Messvektor aktuell nicht verfügbar ist.

Bemerkenswert sind Sternsensoren, welche eine Genauigkeit im Bogensekundenbereich erreichen können. Vor allem bei astronomischen oder Erdbeobachtungsmissionen haben die Teleskope oder Radarnutzlasten ein sehr kleines Sichtfeld von wenigen Grad. Auch bei der High-Speed-Kommunikation in Ka- oder Ku-Band ist die Antennenkeule äußerst schmal. Um zuverlässige Ergebnisse erzielen zu können, muss die Ausrichtung daher extrem präzise sein. Genau hier ist die Lageerfassung mit Sternsensoren unerlässlich. Die meisten kommerziell verfügbaren Sternsensoren sind groß, schwer und verbrauchen mehrere Watt an elektrischer Leistung. Das macht eine Adaption dieser Technologie an die Randbedingungen der Pico- und Nanostellitenmissionen fast unmöglich.

Tabelle 1. Lageregelungssensoren im Vergleich [BAL11]

Sensortyp	Vor- und Nachteile	Typische Daten
Sonnensensoren	Zuverlässig und genau, niedriger Energieverbrauch, freier Blick zur Sonne notwendig	Genauigkeit: $0,01^\circ - 3^\circ$ Masse: 0,025–3 kg Leistungsaufnahme: 0–3 W
Sternsensoren	Große erzielbare Genauigkeit, freies Blickfeld notwendig	Genauigkeit: $0,003^\circ - 0,01^\circ$ Masse: 0,2–8 kg Leistungsaufnahme: 0,5–25 W
Erdhorizontsensoren	Robust und leicht, relativ ungenau	Genauigkeit: $0,01^\circ - 1^\circ$ Masse: 1–5 kg Leistungsaufnahme: 0,5–15 W
Magnetfeldsensoren	Robust und leicht, ungenau	Genauigkeit: $0,5^\circ - 3^\circ$ Masse: 0,06–1 kg Leistungsaufnahme: 0,1–1 W
Solarpaneele	Keine zusätzliche Hardware erforderlich, ungenau	Genauigkeit: 1°
Gyroskope	Sehr genau, aber hoher Kosten-, Massen- und Energiebedarf, Driftphänomene	Genauigkeit: ca. $0,001^\circ$ Masse: 3–25 kg Leistungsaufnahme: 10–200 W

Auf den meisten Pico- und Nanostelliten bzw. CubeSats fliegen entweder keine oder selbstentwickelte Sternsensoren mit. Nur wenige Hersteller (Anhang H) bieten Sternsensoren für Pico- und Nanosatelliten an, die einen Stromverbrauch unter 0,5 W haben und weniger als 150 g wiegen. Laut Statistiken aus dem Jahr 2010 hatten nur ca. 4% der Pico- und Nanosatelliten Sternsensoren an Bord (Abbildung 5).

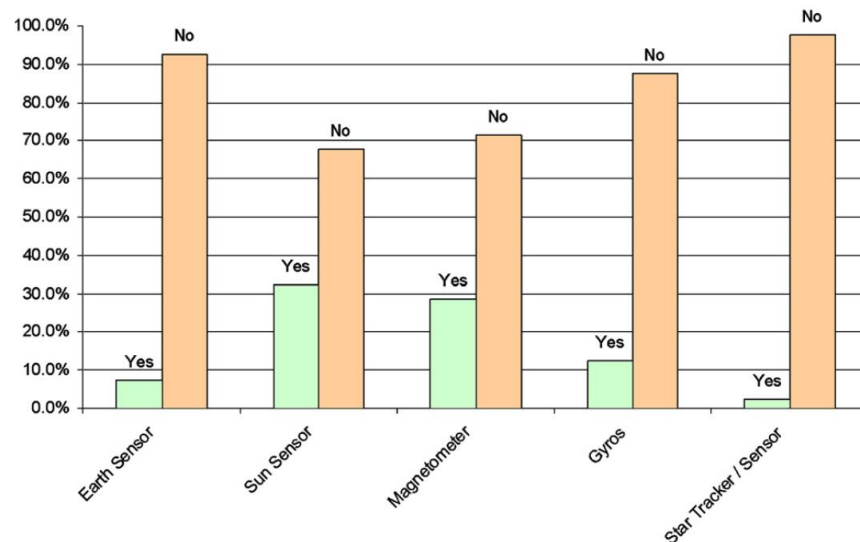


Abbildung 5. Statistiken zur Lageerfassungssensoren [BOU10]

Aktuell haben weniger als ein Viertel aller CubeSats einen oder mehrere Sternsensoren, davon gehören die meisten Satelliten der Planet Labs – Konstellation an. Nach wie vor ist die einfachste und beliebteste Kombination der Lagesensoren: Sonnensensoren mit Magnetometer.

Sternsensoren sind nicht nur die präzisesten, sondern auch die teuersten Lageerfassungssensoren. Die im Anhang H aufgeführten Preise der Hersteller liegen im mittleren fünfstelligen Bereich. Dies ist auf die Komplexität sowie den Entwicklungs- und Betriebsaufwand der Sternsensoren zurückzuführen.

3 Sternsensoren als System

Sternsensoren sind komplexe opto-elektronische Messsysteme. Um die Zusammenhänge und Abläufe in einem Sternsensor besser verstehen zu können, soll hier ein kurzer Blick auf dessen Aufbau und das Funktionsprinzip geworfen werden.

3.1 Aufbau und Funktionsprinzip eines Sternsensors

Ein Sternsensor besteht im Wesentlichen aus drei Hauptkomponenten: Optik, Elektronik mit eingebetteter Software und Struktur. Als Optik wird meistens ein extrem lichtstarkes Refraktor-Objektiv verwendet. Manchmal wird eine Spiegeloptik eingesetzt oder die Optik von einer Nutzlast mitverwendet und der optische Strahl geteilt. Ein Halbleiter-Detektor dient als Schnittstelle zwischen der Optik und der Elektronik. Er wandelt das darauf fallende Licht in ein entsprechendes elektrisches Signal um. Am häufigsten werden CCD oder CMOS Flächen-sensoren mit einer Auflösung von bis zu 2 MPixel verwendet. Die CCD Sensoren liefern Bildinformation noch als analoge Spannung, die in einem nachgeschalteten Analog-Digital-Wandler in ein digitales Signal konvertiert wird. Fast alle CMOS-Sensoren hingegen geben die Bildinformation bereits digital aus und sind mittlerweile kaum in der Empfindlichkeit den CCDs unterlegen. Die Elektronik beinhaltet in der Regel eine oder mehrere Logik-Komponenten, wie z.B. Mikroprozessoren oder FPGAs, die für das Bildauslesen und die Bildverarbeitung zuständig sind. Darüber hinaus ist ein nicht-flüchtiger Speicher für einen Sternkatalog erforderlich. Alle Hardwarekomponenten werden in eine feinmechanische Struktur integriert. Außerdem wird ein Streulichtschutz benötigt, um den störenden Einfluss der direkten oder reflektierten Sonnenstrahlung zu minimieren.

Ein Großteil der Funktionen, die für die Lageerfassung notwendig sind, ist allerdings in der Software verborgen (Abbildung 6). Die Lagebestimmung beginnt mit der Aufnahme eines Sternbildes. In nächstem Schritt werden alle sternähnlichen Objekte anhand spezieller Bildverarbeitungsmethoden aus dem Bild extrahiert und deren Schwerpunkte berechnet. Die aus dem Bild erkannten Sternmuster werden mathematisch beschrieben, um damit dann nach Übereinstimmungen im gesamten Sternkatalog bzw. Datenbank zu suchen, um die Sterne eindeutig zu identifizieren. Die Suche kann mittels der Information aus der vorherigen Lagebestimmung beschleunigt werden. Wurden zwei oder mehr Sterne aus dem Bild erfolgreich dem Sternkatalog zugeordnet, kann die Lage berechnet und in einer gewünschten Darstellungsform ausgegeben werden.

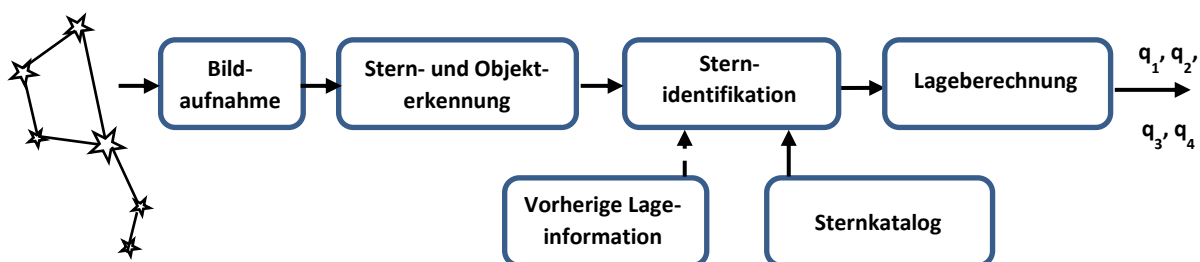


Abbildung 6. Softwarestruktur eines Sternsensors

Wie genau die einzelnen Softwareschritte funktionieren und welche Methoden und Algorithmen dafür am besten geeignet sind, ist Hauptthema des Kapitels 4.

3.2 Motivation und Problemformulierung

Bei der Entwicklung eines Sternsensors für Pico- und Nanosatelliten steht man drei spezifischen Problemen gegenüber, die im weiteren Verlauf dieses Kapitels erläutert werden. Die ersten zwei davon lassen sich auf die Natur des Sternsensorysystems selbst und das physikalische Messprinzip zurückführen. Der dritte Aspekt ist stark an die Betriebsumgebung der Sternsensoren gebunden.

3.2.1 Multiparameter- und multikriterielles Problem in der Entwicklung

Wer sich detailliert mit Entwurf und Konzeption von Sternsensoren beschäftigt, stellt schnell fest, dass die einzelnen Designvariablen eines Sternsensorysystems stark interagieren. Darüber hinaus reagiert das Gesamtsystem unterschiedlich empfindlich auf Änderungen der einzelnen Parameter [BAL11-1].

Abbildung 7 zeigt die Abhängigkeiten zwischen einzelnen Komponenten bzw. Prozessen in einem Sternsensor. Sofort auffallend ist eine zirkuläre Abhängigkeit im Sternsensorysystem. Einer der kritischen Zusammenhänge besteht zwischen Optik, Detektor und Lagebestimmung, dessen konstruktive Auslegung sehr aufwendig und zeitintensiv ist. Brennweite und Öffnung der Optik, Sensorgröße und seine Empfindlichkeit sowie Belichtungszeit bestimmen Anzahl und Position der Sterne im Bild. Die existierenden Algorithmen zur Objekterkennung und Sternidentifikation stellen ihrerseits auch Anforderungen an die minimale Anzahl und Positionsgenauigkeit der Sterne im Sichtfeld. Wenn keine gemeinsame Lösung gefunden wurde, wird weiter an der Verbesserung bzw. der Neuauswahl der Systemkomponenten gearbeitet. Auf Systemebene handelt es sich um ein multidisziplinäres Optimierungsproblem mit mehreren Designvariablen und -zielen, das bei der Entwicklung der Sternsensoren gelöst werden muss.

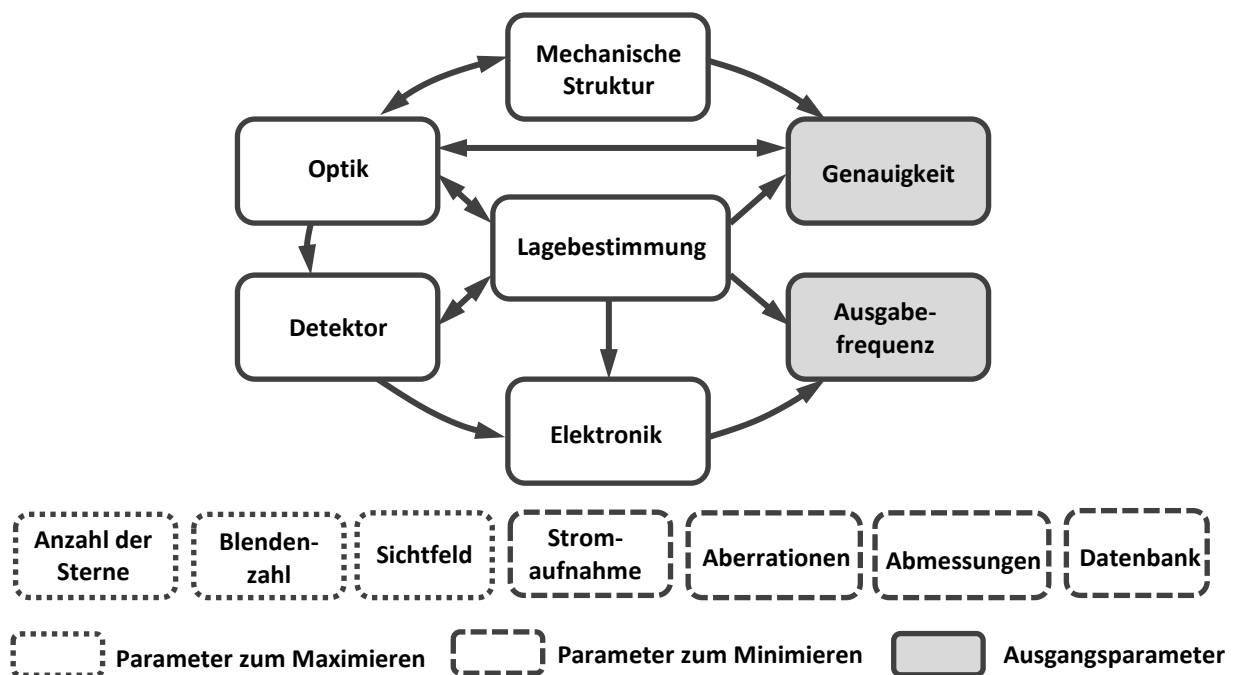


Abbildung 7. Interne Abhängigkeiten in einem Sternsensor

3.2.2 Multidisziplinarität des Designproblems

In der Regel arbeitet eine ganze Gruppe von Ingenieuren und Wissenschaftlern an solchen anspruchsvollen Vorhaben wie einer Sternsensorentwicklung. Unter diesen Umständen wird die gesamte Entwicklungsarbeit in Teilprobleme zerlegt und als Arbeitspakete zwischen den einzelnen Mitarbeiter verteilt. Die Arbeitspakete werden von zuständigen Fachkräften bearbeitet und die Ergebnisse in regelmäßigen Projekttreffen fusioniert und diskutiert.

Im Falle der Sternsensorentwicklung wären beispielweise ein Optikentwickler, ein Entwickler der elektronischen Hardware, ein Software-Entwickler und ein Maschinenbau-Ingenieur erforderlich. Jeder Mitarbeiter versucht, abhängig vom fachspezifischen Hintergrund, sein Arbeitspaket bzw. Teilsystem weitgehend zu optimieren, um eine maximale Performance zu erreichen. Wie Abbildung 8 zeigt, führt maximale Leistung eines Teilsystems nicht unbedingt zur Verbesserung, sondern unter Umständen auch zu einer Verschlechterung der Leistungsfähigkeit des Gesamtsystems. Nur durch ausgesprochen gute Teamarbeit und Kompromissbereitschaft der einzelnen Spezialisten kann die Performance des Gesamtsystems gesteigert werden. Die Entscheidungsgewalt über das endgültige und ausbalancierte Systemdesign liegt in den Händen des Projektleiters bzw. des zuständigen Systemingenieurs. Er hat systemübergreifende Kenntnisse und überwacht, wie erfolgreich die Entwicklung abläuft, ob alle Anforderungen erfüllt sind und die Ziele erreicht wurden [DAE02].

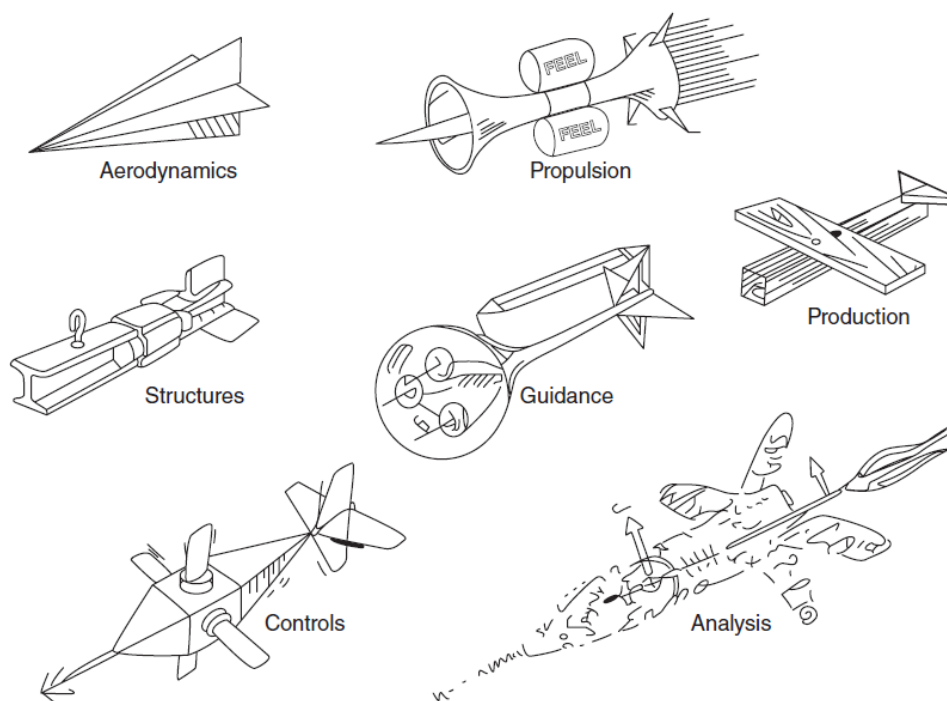


Abbildung 8. Ideales Design einer Rakete aus der Sicht verschiedener Fachexperten [KOS03]

3.2.3 Pico- und Nanosatelliten-Aspekte in der Entwicklung von Sternsensoren

Anforderungen an Sternsensoren kommen an erster Stelle vom ADCS. Dieses wird wiederum durch die Nutzlast bestimmt. Außerdem stellen die Bauform und die Konfiguration eines Satellitenbusses zusätzliche Randbedingungen auf. Deshalb ist die Klassifikation von Sternsensoren etwas abweichend von den Satellitenklassen (siehe Kapitel 2). Im Hinblick auf Pico- und Nanosatelliten werden für die nachfolgende Untersuchung zwei Gruppen von Sternsenso-

ren identifiziert. Zur ersten Gruppe gehören Ultra-Miniatur-Sternsensoren, welche inklusive Baffle weniger als 250 g wiegen und deren größte Abmessung kleiner als 100mm ist. Diese Sternsensoren sind für einen Einsatz in CubeSats von 1U bis 6U gut geeignet. Die zweite Gruppe sind Miniatur-Sternsensoren mit einer Systemmasse pro Kameramodul von 250 g bis etwa 1 kg. Sinnvolle Anwendungen für solche Sensoren finden sich auf CubeSats ab 12U und größer. Bei der Sternsensorklassifikation wurde absichtlich Bezug auf die CubeSats genommen, da die Grenze zwischen den Satellitenklassen etwas verschwommen ist: Obwohl ein 12U CubeSat eine Masse von bis zu 24 kg haben kann, wird er noch zu den Nanosatelliten gezählt. Ein 27U CubeSat ist bereits ein Mikrosatellit und kann unter Umständen bis zu 54 kg schwer sein.

Folglich sind die Randbedingungen für die Ultra-Miniatur-Sternsensoren anders im Vergleich zu den Miniatur-Sternsensoren. Hier dominiert die Kompaktheit des Systems: Abmessungen und Masse sollen möglichst klein gehalten werden. Nutzlast und Hauptkomponenten in einem CubeSat sind äußerst dicht verbaut und Sternsensoren werden da platziert, wo gerade freier Raum vorhanden ist. Die meisten 1U- bis 3U-CubeSats sind Budget-Projekte, weshalb auch ein niedriger Preis essenziell ist. Die Genauigkeitsanforderungen sind hingegen moderat, da die Nutzlasten in der Regel weniger anspruchsvoll sind. Darüber hinaus ist eine sinnvolle mechanische Referenz zwischen den Sternsensoren und der Nutzlast extrem aufwändig und benötigt optische Kalibrierungsvorrichtungen. Diese sind sehr teuer und für kleine Missionen nicht rentabel.

Miniatur-Sternsensoren dürfen etwas größer und schwerer sein. Auch bei der Wahl der Einbaulage ist man flexibler. Die relative Lage zwischen Sternsensoren und der Nutzlast wird meistens vermessen und korrigiert. Der Streulichtschutz, der sogenannte Baffle, ist markanter und aufwändiger. Somit ist der Sonnenausschlusswinkel kleiner. Miniatur-Sternsensoren werden vor allem auf Performance, nämlich Genauigkeit und Ausgaberate, optimiert.

Das Thema „Stromaufnahme“ ist für beide Sternsensorklassen relevant. Bei festmontierten Solarpaneelen werden nur etwa 1,5-2W elektrische Leistung pro 1U in der Sonnenphase generiert. Ein Teil der gewonnenen Energie wird direkt vom Satellitenbus verbraucht, der Rest wird für die Schattenphase gespeichert oder für die Experimente verwendet. Im nominalen Betrieb verbrauchen die kommerziellen Ultra-Miniatur-Sternsensoren durchschnittlich 0,5W, die größeren Varianten sogar bis zu 3W. Der hohe Stromverbrauch der Sternsensoren im Vergleich zu den übrigen ADCS-Sensoren beschränkt stark deren operationelle Zeit.

Der Einfluss der Anforderungen und Randbedingungen von Pico- und Nanosatelliten auf das Design von Sternsensoren wird in Kapitel 7.2. noch genauer untersucht.

4 Besonderheiten der Entwicklungsprozesse in der Raumfahrt

Die Raumfahrt als Industriebranche ist sehr konservativ. Alle Abläufe haben eine vorgegebene und standardisierte Struktur und werden streng kontrolliert. Abhängig vom Staat bzw. der Raumfahrtagentur sind unterschiedliche Raumfahrtstandards gültig. Diese haben aber grundsätzlich sehr ähnliche Inhalte. Im europäischen Raum sind die ECSS-Standards gebräuchlich. Der Entwicklungsablauf eines Raumfahrtsystems ist z.B. im Standard ECSS-M-ST-10C „Space project management, project planning and implementation“ genau beschrieben [ECS09].

4.1 Entwicklung nach Projektphasen

Ein Raumfahrtprojekt wird typischerweise in sieben Projektphasen unterteilt. Jede Projektphase endet mit einem Meilenstein in Form eines bestimmten Reviews [AGU13]. Im Review werden die Ergebnisse der gesamten Phase überprüft und eine Erlaubnis für den Übergang zur nächsten Phase erteilt.

Phase 0: Missionsanalyse / Identifizierung der Erfordernisse. Jede Mission beginnt mit der Erstellung des Leitbildes. Anhand identifizierter Missionsbedürfnisse, erwarteter Leistung sowie Zuverlässigkeits- und Sicherheitszielen wird die geplante Mission charakterisiert. Auch Randbedingungen hinsichtlich der physischen Betriebsumgebung werden analysiert. Es entstehen die ersten vorläufigen Anforderungskataloge. Mögliche Missionskonzepte auf Top-Level werden ausgearbeitet. Anschließend werden erste Markt- und Risikoanalysen durchgeführt. Seitens des Projektmanagements entstehen Pläne bezüglich der Projektorganisation, der Kosten und des Zeitaufwandes. Am Ende dieser Phase findet ein Missionsdefinitions-Review (MDR) statt.

Phase A: Durchführbarkeit. In dieser Projektphase werden erste Untersuchungen zu möglichen System- und Betriebskonzepten sowie zu Systemarchitekturen durchgeführt. Unter Berücksichtigung der identifizierten Erfordernisse werden Projektunsicherheiten und Risiken sowie kritische Elemente für jedes Konzept bestimmt. Es entstehen Studien zur technischen und programmatischen Machbarkeit der möglichen Konzepte unter Berücksichtigung der Randbedingungen hinsichtlich Umsetzung, Kosten, Zeitplan, Organisation, Betrieb, Produktion und Entsorgung. Kritische Technologien und Elemente werden identifiziert und quantitativ beschrieben. Vorschläge zum System- und Betriebskonzept sowie der technischen Lösungen, einschließlich Modellphilosophie und Verifizierungsansatz, werden ausgearbeitet. Die vorläufigen Anforderungskataloge sowie Projektmanagementpläne aus der Phase 0 werden vervollständigt und freigegeben. Die Phase endet mit dem vorläufigen Anforderungs-Review (PRR).

Phase B: Vordefinition. An erster Stelle wird die Durchführbarkeit der technischen Lösungen für die System- und Betriebskonzepte unter Berücksichtigung der programmatischen Randbedingungen bestätigt. Es werden "Trade-off"-Studien durchgeführt und ein bevorzugtes Systemkonzept ausgewählt. Die benötigten Systemeigenschaften und Funktionalitäten werden als Systemanforderungen zusammengefasst. Eine vorläufige Designdefinition für das ausgewählte Systemkonzept und die relevanten technischen Lösungen wird erstellt. Es entstehen

ein Produkt- und Spezifikationsbaum sowie Projektstrukturplan. Erste Untersuchungen von "Eigenfertigung oder Kauf"-Alternativen werden durchgeführt. Die Beschaffung von Komponenten mit langer Lieferzeit, die zum Einhalten des Projektzeitplans erforderlich sind, wird initiiert. Es finden auch erste Vorentwicklungsarbeiten an kritischen Technologien oder in Systemdesignbereichen statt, in denen es notwendig ist, die Entwicklungsrisiken zu reduzieren. Auch ein Verifikationsprogramm inklusive Modellphilosophie wird erstellt. Dokumente zum Projektmanagement wie Zeit- und Kostenplan sowie Risikobewertung werden vervollständigt und finalisiert. In Projektphase B werden normalerweise zwei Reviews durchgeführt: das Systemanforderungs-Review (SRR) und zum Schluss ein vorläufiges Design-Review (PDR).

Phase C: Detaildefinition. Das Ziel dieser Projektphase ist die Fertigstellung der detaillierten Designdefinition des Systems auf allen Ebenen. Alle externen und internen Schnittstellen des Gesamtsystems werden ausführlich definiert und in entsprechenden Schnittstellen-Kontroll-Dokumenten (ICDs) und Handbüchern festgehalten. Die ausgewählten, kritischen und relevanten Elemente sowie Komponenten werden produziert und getestet. Abhängig von der gewählten Modellphilosophie werden anschließend benötigte Modelle der einzelnen Komponenten und des Gesamtsystems hergestellt und Entwicklungstests durchgeführt. Es werden konkrete Pläne zur Montage, Integration und Verifikation ausgearbeitet. Die Phase endet mit einem kritischen Design-Review (CDR).

Phase D: Produktion/Bodenqualifikationstests. Die Entwicklung des Komplettsystems wird vollständig abgeschlossen. Die geplanten Qualifizierungstests und alle zugehörige Verifizierungsaktivitäten werden durchgeführt. Wenn die Tests erfolgreich durchlaufen wurden, wird die Flug-Hardware in Produktion gegeben. Nach der Fertigstellung der finalen Flughardware werden Akzeptanztests vorgenommen. Auch die bodenunterstützende Ausrüstung wird fertiggestellt. Es laufen finale Kompatibilitätstests zwischen Raum- und Bodensegment. Diese Phase enthält zwei Reviews: Qualifikations-Review (QR) und Abnahme-Review (AR).

Phase E: Betrieb. Diese Phase umfasst die Startkampagne, die LEOP (Launch and Early Orbit Phase) und die Inbetriebnahme sowie den Routinebetrieb des Produkts. Im Laufe dieser Phase müssen alle Aktivitäten zur Startvorbereitung, der Start selbst, die Inbetriebnahme im Orbit und anschließend der reguläre Betrieb durchgeführt werden. Während dieser Phase finden diverse Reviews statt. Die wichtigsten sind Flugbereitschaft-Review (FRR), Betriebsbereitschaft-Review (ORR) und Inbetriebnahme-Review (CRR).

Phase F: Entsorgung. Diese Phase signalisiert das Ende der Mission. Der Entsorgungsplan wird umgesetzt. Das Missions-Abschluss-Review (MCR) findet am Ende dieser Phase statt.

Das Einhalten aller ECSS-Standards ist sehr aufwändig und zeitintensiv und daher nur für große Missionen mit dutzenden Mitarbeiter möglich. Die meisten Pico- und Nanosatelliten werden lediglich angelehnt an die ECSS-Standards entwickelt.

4.2 Besonderheiten bei Pico- und Nanosatellitenprojekten

Die meisten Pico- und Nanosatellitenmissionen haben zwei charakteristische Randbedingungen: straffe und sehr oft fixe Zeit- und Kostenpläne. Diese beiden Randbedingungen müssen

gemeinsam betrachtet werden, weil sie sich gegenseitig stark beeinflussen. Die Gesamtmissionskosten umfassen Entwicklungskosten, Startkosten und Betriebskosten. Etwa ein Drittel der Entwicklungskosten sind hardwarebezogene Kosten für den Satellitenbus. Die übrigen zwei Drittel sind Personalausgaben, welche proportional zur Entwicklungsdauer und zur Anzahl der Projektmitarbeiter sind. Die Entwicklung eines universitären CubeSats dauert typischerweise 18-36 Monate, abhängig von der Größe des Projektteams [NAS17, RIC15]. Die Startkosten stehen im direkten Zusammenhang mit der Satellitenmasse und dem benötigten Betreuungsaufwand am Startplatz. Deshalb sind etwa 93% aller Pico- und Nanosatelliten im CubeSat Format ausgeführt [VIL19, NAN19]. Diese fliegen kostengünstig als zweitrangige Passagiere mit und benutzen Standardcontainer für Separation.

Wie das Teufelsquadrat von Harry M. Sneed (Abbildung 9) zeigt, ist der Schwerpunkt von CubeSat-Projekten im Vergleich zu konventionellen Raumfahrtprojekten nach ECSS stark in Richtung „Kosten-Dauer“ verschoben.

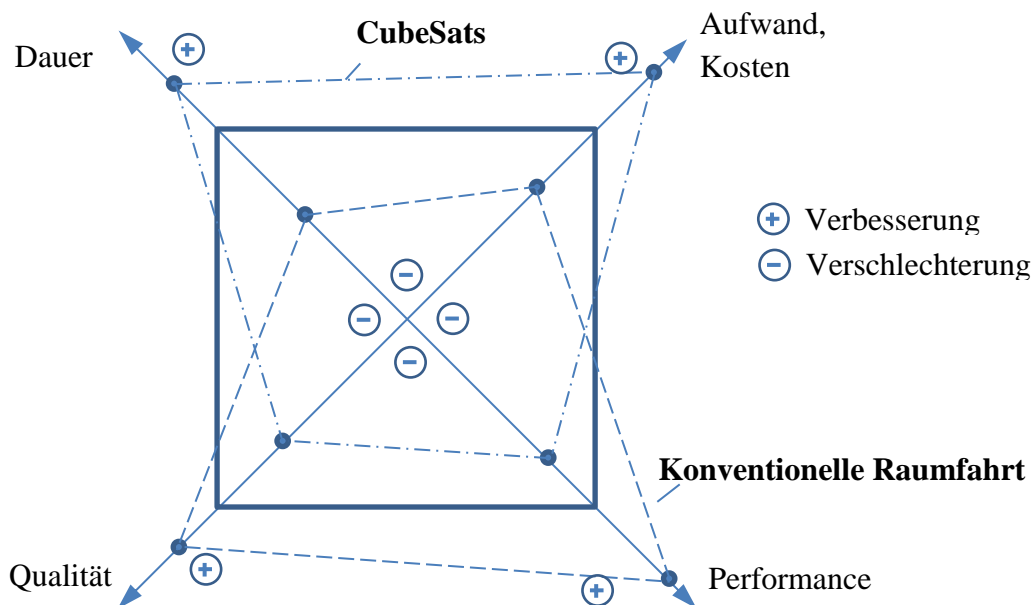


Abbildung 9. CubeSats vs. ECSS basierten Raumfahrtvorhaben

Durch kurze Entwicklungszeit und ein kleines Budget sinken die Qualität der Arbeiten und die Leistungsfähigkeit des Satellitensystems. Dadurch steigt auch das Risiko für ein Scheitern der Mission.

Noch immer gelten die Entwicklungen im Bereich der Pico- und Nanosatelliten als sehr risikobehaftet. Zunächst sind die Projektlaufzeiten relativ kurz. Manchmal wird das Verifikationsprogramm gekürzt und beispielweise Strahlungs- und Schocktests vernachlässigt. Diese sind aber essenziell, wenn man mit COTS-Komponenten (commercial off-the-shelf) arbeitet, die nicht für den Einsatz im Weltall konzipiert sind. Es fehlt auch häufig die Zeit, um zeitintensive End-to-End-Tests mit dem Komplettsystem inklusive Bodensegment durchführen zu können. Ein weiterer Aspekt ist, dass die meisten CubeSats in einer universitären Umgebung bzw. an kleinen Forschungseinrichtungen gebaut werden, wo die Projektbeteiligten, vorwiegend junge wissenschaftliche Mitarbeiter, Doktoranten oder Studenten, keine Erfahrungen in

der Entwicklung von Satelliten haben. Etwa die Hälfte solcher Missionen scheitert bereits in der LEOP. Dagegen liegt die Erfolgsrate bei erfahrenen Entwicklern, die bereits am Bau eines oder mehrerer Satelliten beteiligt waren, etwa doppelt so hoch [SWA19].

4.3 Top-Down-Verfahren

Bei der Entwicklung eines Raumfahrtssystems empfehlen die ECSS-Standards das V-Vorgehensmodell zu nutzen (Abbildung 10). Auch die bereits vorgestellten Projektphasen und Meilensteine in Form von Reviews spiegeln diese Vorgehensweise wieder. Der gesamte Prozess ist dabei in zwei Hälften geteilt: Systementwurf und Implementierung / Tests.

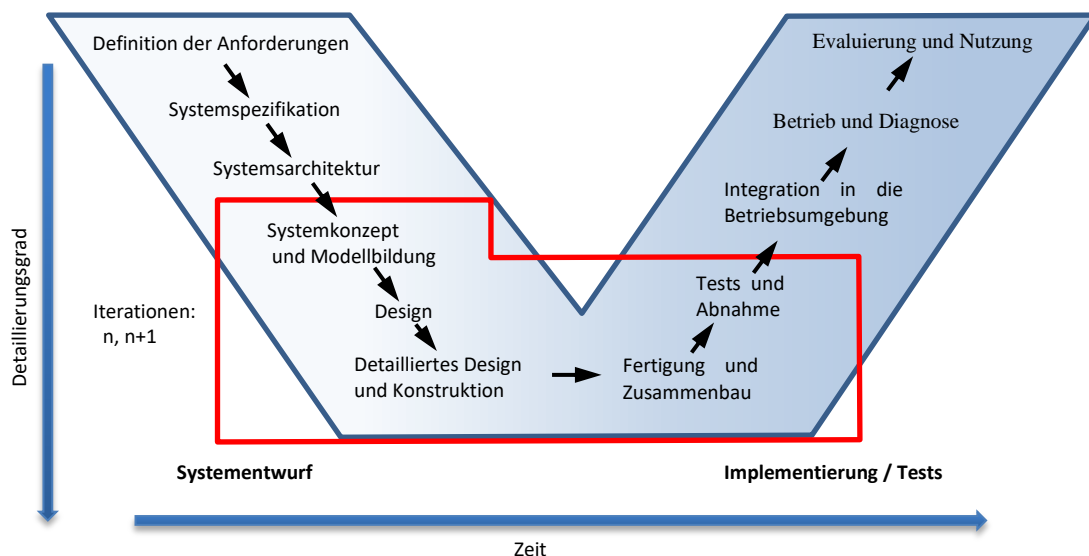


Abbildung 10. V-Modell der Systementwicklung [angelehnt an AGU13]

Die Entwicklung einer Raumfahrtkomponente oder eines Systems beginnt mit Erstellung der Systemanforderungen und anschließend einer Systemspezifikation, die aus übergeordneten Missions- und Nutzeranforderungen sowie Zielen abgeleitet werden. Anhand dieser Spezifikation werden eine oder mehrere Systemarchitekturen und -konzepte erstellt, bewertet und ein konkreter Designvorschlag ausgearbeitet. Im nächsten Schritt wird das ausgewählte Design vertieft und die einzelnen Bestandteile im Detail entworfen. Der Implementierungsteil beginnt mit der Fertigung und dem Zusammenbau der ersten Muster. Diese werden dann diversen Tests unterzogen. Im Falle positiver Testergebnisse werden die dafür vorgesehenen Systemmodelle unter relevanten Bedingungen in Betrieb genommen und eine Langzeitevaluierung durchgeführt.

Falls das gewählte Konzept in der Testphase der kritischen Systemkomponenten scheitert, wird ein neues Konzept ausgewählt und eine neue Entwicklungsiteration vorgenommen. Leider sind solche Iterationen im V-Modell nicht direkt sichtbar und für die Entwicklung eines komplett neuartigen Systems schlecht kalkulierbar. Nur Entwickler mit viel Erfahrung können sich anbahnende Designprobleme frühzeitig erkennen und diese schnell lösen. Dadurch werden Iterationen vermieden bzw. kurz gehalten. Alternativ kann an mehreren Lösungen parallel gearbeitet werden. Am Ende werden dann die Ergebnisse fusioniert. Dies ist jedoch sehr personalintensiv.

4.4 Iterative Entwicklung

Ein realistischeres Szenario bei Entwicklungen im Pico- und Nanosatellitenbereich beschreibt das Spiral-Modell. In diesem Modell sind einzelne Iterationen deutlich dargestellt. Die Entwicklungsphasen A bis C werden solange auf Material-, Komponenten- bzw. Systemebene wiederholt, bis die gestellten Anforderungen erfüllt sind.

In jeder Iteration der Entwicklung muss eine erneute Zeit- und Budgetplanung vorgenommen werden. Es muss gründlich analysiert werden, ob eine weitere Iteration die Projekttrandbedingungen verletzt oder die vorhandene, aber vielleicht suboptimale Lösung zum Teilerfolg führen kann.

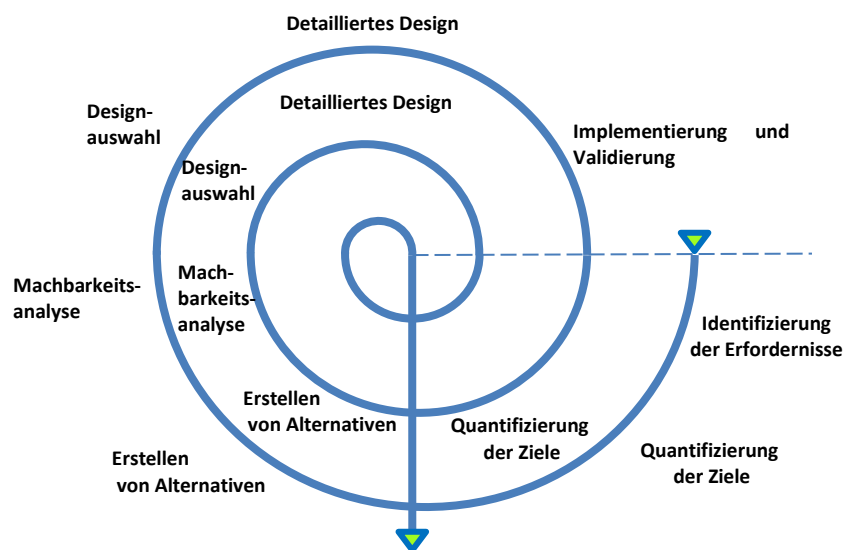


Abbildung 11. Spiral-Modell der Entwicklung [AGU13]

Prinzipiell enthält jede Entwicklung kleine oder große Iterationen bzw. Schleifen. Diese sind immer mit einem Zeit- oder Personalaufwand verbunden und sollten vor allem in kleinen Vorhaben, wie CubeSats, minimiert werden. Wichtig ist, dass die Endlösung zielführend ist. Ein Beispiel einer erfolgreichen iterativen Entwicklung eines Sternsensors liefert das Vorhaben STELLA.

4.5 STELLA Sternsensor als Beispiel der iterativen Entwicklung

STELLA ist ein Sternsensor für Pico- und Nanosatelliten, der an der Professur für Raumfahrttechnik der Universität Würzburg entwickelt wurde. Der Vorhaben wurde vom Bundesministerium für Wirtschaft und Energie unter dem Förderkennzeichen 50RM0901 finanziert. Hauptziel des Vorhabens war es die vorhandenen Sternsensortechnologien zu untersuchen und zu miniaturisieren. Dabei sollten die Randbedingungen der erwähnten Satellitenklassen bezüglich Bauvolumen, Masse und Stromverbrauch berücksichtigt werden. Darüber hinaus sollten Sternsensoren gebaut und für einen Raumfahreinsatz qualifiziert werden [KAY09, WOJ11].

Die Entwicklung wurde innerhalb von drei Jahren zwischen 2009 und 2012 vollständig abgeschlossen. Es wurden fünf Modelle gebaut, qualifiziert und anschließend unter realem Sternhimmel auf Funktion getestet. Die Arbeiten wurden durch zwei wissenschaftliche Mitar-

beiter ausgeführt. Der Autor dieser Arbeit übernahm hierbei den Systementwurf, die eingebettete Implementierung der Sternsensoralgorithmen, Optik- und Strukturdesign sowie Integration und Tests. Herr Harald Wojtkowiak war für das Design der elektronischen Hardware und die FPGA-Programmierung in VHDL sowie die Schnittstellenimplementierung zuständig.

Systemübersicht

Der STELLA-Sternsensor (Abbildung 12) war zum Zeitpunkt des Entwicklungsendes einer der kleinsten und energieeffizientesten Sensoren seiner Klasse. Der Sensor kann eine vollständige Lagebestimmung innerhalb von einer Viertelsekunde durchführen und verbraucht dabei nur 250 mW an elektrischer Leistung. Die vollständige Spezifikation ist Tabelle 2 zu entnehmen [BAL12].

Systemdesign

Beim Design des Sternsensors wurden die Empfehlungen des ECSS weitestgehend beachtet. Nach dem Erstellen der Anforderungen und Systemspezifikationen wurde die kritischste Technologie untersucht. Als kritischste Komponente wurden die CMOS-Detektoren identifiziert. Zu diesem Zeitpunkt gab es nur wenige Hersteller, die CMOS-Flächensensoren für den offenen Markt produzierten. Eine weitere Problematik war die Anschaffung der Detektoren in kleinen Mengen.

Tabelle 2. Spezifikation STELLA

Parameter	Wert
Genauigkeit	0,01° pitch/yaw, 0,04° roll 3 σ
Sichtfeld	14,3° diagonal
Stromversorgung	+3,3 V
Leistungsaufnahme, Peak	250 mW
Abmessungen inkl. Baffle	91 x 46 x 58 mm ³
Masse inkl. Baffle	170 g
Betriebstemperatur	-25...+55°C
Lebensdauer im LEO	2 Jahre (Ref.-Orbit 500 km)
Ausgabeformat	Quaternion
Ausgaberate	4 Hz
Schnittstellen	2x CAN 2.0B, 2x UART



Abbildung 12. STELLA EQM Modell

In der ersten Iteration wurde der Sensor IBIS5 des Herstellers Cypress Semiconductor ausgewählt. Der Hersteller versprach sehr rauscharme Aufnahmen mit einer damals ordentlichen Empfindlichkeit von etwa 3,5 V/lux·s. Die weitere Entwicklung verlief entsprechend dem Diagramm in Abbildung 13. Nach der Selektion des Detektors wurden Brennweite und Öffnung des optischen Systems bestimmt, um bei einer vorgegebenen Belichtungszeit die benötigte Anzahl an Sternen im Sichtfeld (in der Regel größer als 3) zu erhalten. Danach wurde ein Identifikationsalgorithmus ausgewählt und die Größe des entsprechenden Katalogs bestimmt. Um eine Ausgaberate von etwa 4 Hz zu erreichen, wurde ein passender FPGA-Chip und Digi-

tal-Signal-Prozessor sowie Speicherbausteine ausgesucht. Auf Basis dieser Komponenten wurde ein Engineering-Modell entworfen und hergestellt. Die Feldtests am realen Sternenhimmel zeigten aber, dass der IBIS5-Sensor nur ein Viertel der im Datenblatt angegebenen Empfindlichkeit aufweist und somit nicht die erwartete Sternenaufnahme liefert.

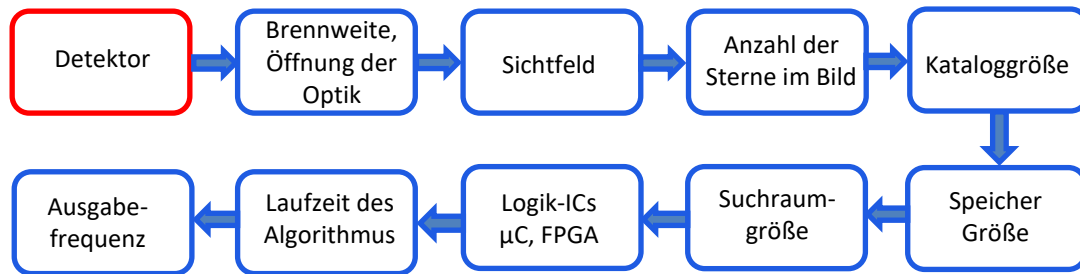


Abbildung 13. Entwicklungsablauf des STELLA-Sternsensors

Für die nächste Entwicklungsiteration wurde ein neuer Detektor vom europäischen Hersteller E2V gewählt. Das Sternensensorsystem musste erneut für den E2V-Bildsensor ausgelegt werden [BAL11-1].

Hardware

STELLA hat eine für einen Sternsensor klassische Systemarchitektur und besteht aus folgenden Komponenten: Elektronik, Optik, Gehäuse mit Baffle und Software. Alle Komponenten außer der Optik sind eine Eigenentwicklung der Professur für Raumfahrttechnik der Universität Würzburg.

Elektronische Hardware

Die elektronische Hardware des STELLA-Sternsensors besteht aus drei separaten Leiterplatten (Abbildung 14). Jede Leiterplatte übernimmt eine bestimmte Anzahl an in sich abgeschlossenen Aufgaben. Das Sensor-Board ist für die Aufnahme und temporäre Speicherung des Bildes zuständig. Auf der Vorderseite dieser Leiterplatte befindet sich ein monochromer Bildsensor mit einer Auflösung von 860 x 640 Pixeln und auf der Rückseite ein ProASIC3 FPGA-Chip. Dieser steuert den Bildsensor an und speichert das aufgenommene Bild in einem 2 MB großen SRAM-Speicherbaustein ab. Der Zugriff auf den SRAM-Speicher ist zwischen dem FPGA und dem DSP C5000 des Prozessor-Boards geteilt. Wenn die Aufnahme eines Bildes abgeschlossen ist, fängt der DSP mit der Bildverarbeitung an und extrahiert die Sterne aus der Aufnahme. Im nächsten Schritt wird das aus den erkannten Sternen gebildete Muster geometrisch beschrieben. Diese Beschreibung wird mit den Einträgen im Katalog verglichen und es wird nach Übereinstimmungen gesucht. Der Katalog ist nicht-flüchtig in einem 8 MB großen Flash-Speicher abgelegt. Wenn die Identifikation von mindestens zwei Sternen erfolgreich war, kann der DSP die Lage berechnen.

Der STELLA-Sternsensor benutzt keine rekursiven Informationen aus der vorangegangenen Berechnung. Die Lage wird aus jeder Sternenaufnahme neu berechnet. Die berechnete Lage wird dem Service-Board über einen I2C-Bus mitgeteilt. Dieses Board stellt die Schnittstelle zur Außenwelt dar. Zwei redundante AVR-Mikrokontroller auf diesem Board kommunizieren über CAN2.0B mit dem OBC- oder ADCS-Computer und führen die empfangenen Tele-

kommandos aus. Außerdem sind sie für Erfassung und Verwaltung der Housekeepingdaten zuständig. Weitere Funktionen dieser Leiterplatte sind die Aufbereitung der benötigten Spannungen und die Steuerung der Schaltzustände des Sensor- und Prozessorboards. Auf dem Service-Board werden auch alle Softwareimages in einem SPI-Flash gespeichert und verwaltet.

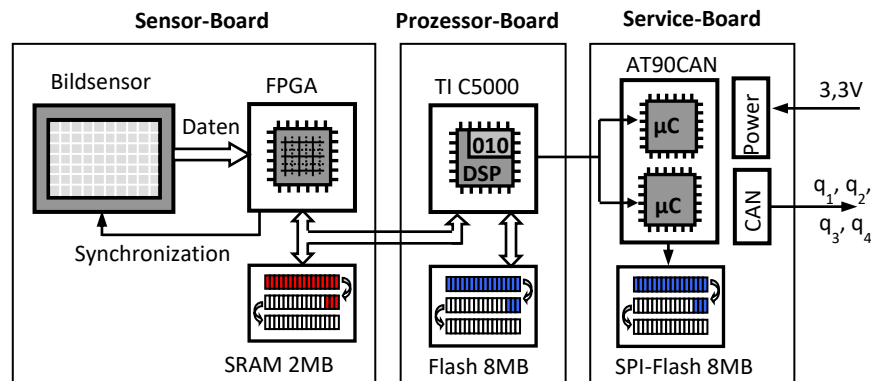


Abbildung 14. Architektur der Elektronik vom STELLA-Sternsensor

Alle drei Boards sind 40 mm x 40 mm groß und sind in einem Stapel angeordnet. Die elektrische Verbindung zwischen den benachbarten Boards erfolgt über 40-polige COTS Steckverbinder. Für die externe Anbindung des gesamten Sternsensors wurde ein raumfahrtqualifizierter 21-poliger Nano-D Stecker eingesetzt.

Optisches System

Die Hauptaufgabe des optischen Systems ist es die Sterne möglichst ohne Verzerrungen auf den Bildsensor zu projizieren. Für den STELLA-Sternsensor wurde ein eigenes lichtstarkes Refraktor-Objektiv entwickelt und hergestellt. Um die Startbelastungen gut überstehen zu können, wurde das Design auf Schock- und Vibrationsfestigkeit optimiert. Außerdem wurden konstruktive Maßnahmen zur schnellen Entlüftung der Linsenzwischenräume im Vakuum durchgeführt. Das Objektiv besteht aus fünf einzelnen Linsen, die gesondert eingefasst und zentriert wurden. Die Brennweite des Objektivs beträgt 25 mm und die Öffnung $f/1,6$. In Kombination mit dem Bildsensor ergibt sich ein relativ schmales Sichtfeld von $11,5^\circ \times 8,6^\circ$. Bei einer nominalen Belichtungszeit von etwa 100 ms können Sterne mit der Magnitude bis zu 6,7 beobachtet werden.

Mechanische Struktur

Die mechanische Struktur des Sensors ist nicht nur für die Akkommodation aller Komponenten zuständig, sondern dient auch als mechanische Referenz für das Messsystem. Die optische Achse des Objektivs, der Bildsensor und die Auflageflächen der Sternsensorbefestigung müssen sehr präzise zueinander ausgerichtet sein. Andernfalls wird die Genauigkeit der Lagebestimmung signifikant beeinträchtigt.

Eine weitere Aufgabe der Struktur ist es unter mechanischer Belastung für die Integrität des Gesamtsystems zu sorgen. Die Leiterplatten des STELLA-Sternsensors sind als Stapel angeordnet und jeweils mit einem Strukturteil eingerahmt (Abbildung 15).

Alle Gehäuseteile inklusive Streulichtschutz sind in Raumfahrtaluminium 6061-T6 ausgeführt und in schwarz-matt harteloxiert, um störende Reflexionen der Sonnenstrahlung zu vermeiden.



Abbildung 15. Mechanischer Aufbau des STELLA-Sternsensors

Software und Algorithmen

Die Software des STELLA-Sternsensors lässt sich in zwei Gruppen unterteilen: Lagerfassungssoftware, welche alle benötigten Schritte durchführt, um die Lage aus einem Sternmuster zu errechnen; und Dienstsoftware, die für das Sammeln der Telemetrie zuständig ist und das System parametrieren. Alle Prozesse zur Bildaufnahme und -speicherung wurden in VHDL im FPGA implementiert. Der DSP übernimmt Bildverarbeitung mit Objekterkennung, Sternidentifikation und Lageberechnung. Um die Sterne in einem aufgenommenen Bild erkennen zu können, wird eine Variante der sequenziellen Regionenmarkierung eingesetzt. Bei der Sternidentifikation wird ein innovativer Kreis-Muster-Algorithmus verwendet. Die Lage wird anhand der QUEST-Methode bestimmt. Ein Großteil der Dienstsoftware läuft auf einem der beiden kaltredundanten AVR-Mikrocontroller von Atmel. Der Programcode des DSPs und der Mikrocontroller wurde in C geschrieben.

Ergebnisse und Optimierungspotential

Die gefertigten Modelle wurden zuerst einem Qualifikationsprogramm unterzogen. Das Programm bestand aus einem Strahlungstest mit einer Co60-Quelle bis zu einer Gesamtionisationsdosis von 10 kRad, mechanischen Belastungstests und Thermal-Vakuum-Tests. Alle Tests wurden nach den Spezifikationen der ECSS durchgeführt. Die mechanischen Tests beinhalteten Sinus- und Random-Belastungen sowie Pyroschock-Tests. Bei den Thermal-Vakuum-Tests wurde die Weltraumumgebung inklusive Startprozedur durch Thermalschock und acht weiteren Temperaturzyklen von -35°C bis $+65^{\circ}\text{C}$ im Hochvakuum simuliert. Alle Qualifikationstests verliefen erfolgreich [BAL11-2].

Das Bodenverifikationsprogramm wurde mit funktionalen Feldtests am realen Sternenhimmel abgeschlossen. Als Standort wurde hierfür das Kooperationsgelände der Universität am Roque-de-los-Muchachos-Observatorium auf La Palma, Spanien gewählt. Das Gelände befindet sich in 2400 m Höhe und zeichnet sich durch besonders geringe Lichtverschmutzung und günstige Wetterverhältnisse aus.

Der STELLA-Sternsensor wurde im Jahre 2017 an Bord des Satelliten TechnoSat der TU Berlin in einem 600 km Orbit erprobt und führte eine erfolgreiche Technologiedemonstration durch [BAR13]. Während mehrerer Experimente im Weltraum lieferte das System Lageinformationen.

STELLA ist nach wie vor einer der energieeffizientesten Sternsensoren. Vom Bauvolumen her nimmt der Sternsensor etwa ein Viertel eines 1U-CubeSats ein. In redundanter Ausführung ist STELLA somit für den operationalen Betrieb in Satelliten ab 15kg bzw. 12U optimal geeignet.

Um die vorgestellten Technologien in der Lageregelung von Pico- und Nanosatelliten aktiv einsetzen zu können, ist eine weitere Miniaturisierung notwendig. Eine weitere Systemoptimierung hinsichtlich Abmessungen und Masse ist mit großem Aufwand verbunden, vor allem im Bereich Optik und Sensorik. An dieser Stelle sind noch mehrere Entwicklungsprobleme in weiteren Iterationen zu bewältigen.

4.6 Ziele der Arbeit

Die Erfahrungen aus dem STELLA-Vorhaben zeigen, dass das herkömmliche iterative Ablaufmodell in der Entwicklung der Sternsensoren sehr zeitintensiv ist und unter Berücksichtigung der kurzen Projektlaufzeiten der Pico- und Nanomissionen nur zu einer suboptimalen Lösung führt. Deshalb soll ein anderes, effektiveres Verfahren für die Sternsensorentwicklung eingesetzt werden.

Das Ziel dieser Arbeit ist die Entwicklung und Implementierung einer neuen automatischen Designstrategie, welche die klassenspezifischen Anforderungen und Randbedingungen von Pico- und Nanosatelliten berücksichtigt.

Alle Designabläufe sollen computerbasiert sein und automatisiert werden (Computer Aided Engineering, CAE), um die Beteiligung eines Entwicklers zu minimieren. Dies bedeutet, dass alle Berechnungen und der Informationsaustausch zentralisiert in gleicher Umgebung auf einem oder mehreren Rechnern ablaufen sollen. Die Variantenbildung, die Auswertung der Designs und die Entscheidungsfindung über die Notwendigkeit von Änderungen und weiteren Iterationen sollen automatisch ablaufen. Diese Maßnahmen sollen die Entwicklung beschleunigen und den zuständigen Systemingenieur der Sternsensoren entlasten.

5 Modellbasierter Entwurf eines optimalen Sternsensorsystems

In der Zielsetzung der Arbeit wurde die CAE-Fähigkeit der Entwicklungsstrategie als eine der Hauptanforderungen vorausgesetzt. CAE steht im direkten Zusammenhang mit modellbasierter Entwicklungsstrategie.

5.1 Modellbasierte und optimierende Entwicklungsstrategie

Wie zuvor erwähnt, hat die Entwicklung eines Sternsensors einen multidisziplinären Charakter und ähnelt damit dem Designprozess von mechatronischen Systemen. Allerdings enthält ein Sternsensorsystem nicht nur Komponenten aus den Bereichen Mechanik, Elektronik und Informatik, sondern auch der Optik. Abhängig von der Disziplin sind eigene Vorgehensstrategien in der Entwicklung etabliert und werden effektiv eingesetzt. Die Verwendung von spezifischen Strategien schränkt jedoch die Multidisziplinarität des Entwicklungsablaufs ein. Eine der gut bewährten und disziplinunabhängigen Methoden ist die **modellbasierte Systementwicklung** [EIG14]. Diese nutzt eine mathematische Beschreibung des Gesamtsystems und der relevanten Betriebs- bzw. Testumgebung, um Systemverhalten computergestützt zu simulieren. Die Ergebnisse der Simulation unterstützen die Entwicklung in allen Projektphasen, beginnend von Anforderungserfassung bis zur Evaluierung und Nutzung. Besonders in der frühen Konzeptphase hat die modellbasierte Systementwicklung einen enormen Vorteil: sie hilft bei der Bewertung der ausgearbeiteten Systemkonzepte. Anhand der errechneten Leistungsparameter wird dann ein primäres Konzept ausgewählt und weiterverfolgt. Wenn sogar in der Validierungsphase das Konzept fehlschlägt, kann eine neue Entwicklungsiteration schneller erneut modelliert werden.

Bei der Modellbildung geht es nicht um CAD-Tools, die einzelne Komponenten des Systems beschreiben, sondern um ein zentrales Gesamtmodell, das alle wichtigen Zusammenhänge eindeutig wiedergibt, konsistent und berechenbar ist. Die Erstellung eines solchen Systemmodells ist sehr aufwendig und rentiert sich nur, wenn dieses mehrmals und langjährig in verschiedenen Entwicklungen benutzt wird.

Der modellbasierte Ansatz ist nur ein ressourcensparendes Hilfsmittel und garantiert nicht, dass eine optimale Systemlösung gefunden wird. Es ist die Aufgabe des Entwicklers, das Modell so zu variieren, dass eine Verbesserung der Systemleistungsfähigkeit in jeder Designiteration erzielt wird. Wenn die Systemkomponenten einen sehr komplexen Zusammenhang haben, wie es bei den Sternsensoren der Fall ist, kann es unter Umständen schwierig sein, eine geeignete Konfiguration für eine bessere Lösung zu finden. Die möglichen Suchstrategien aus der Sicht des Systems-Engineering sind in Abbildung 16 zusammengefasst.

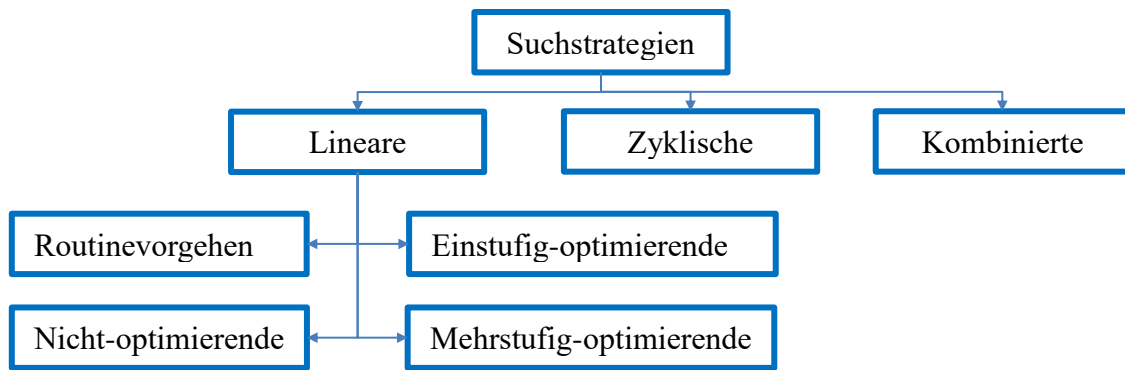


Abbildung 16. Mögliche Strategien zur Lösungssuche [HAB99]

Lineare Suchmethode enthalten keine Schleifen oder Iterationen. Das Routinevorgehen ist die einfachste Variante und ist nur für Designprobleme geeignet, wo die Lösung offensichtlich ist. Die einzelnen Vorgehensschritte folgen direkt nacheinander. Es wird ein klares Designkonzept ohne jegliche Art von Abzweigungen oder Auswahlproblemen verfolgt. Bei der nicht-optimierende Suche hingegen muss in jedem Teilschritt nach einer möglichen funktionsfähigen Lösung gesucht werden. Sobald eine Lösung gefunden wurde, geht man zum nächsten Schritt über. Im Unterschied dazu werden bei der einstufig-optimierenden Vorgehensweise in jedem Vorgehensschritt verschiedene Lösungsvarianten gebildet und anschließend bewertet. Mit der besten Lösung wird dann im nächsten Schritt weitergearbeitet. Bei dieser Methode bleibt man oft in lokalen Maxima auf Komponenten bzw. Subsystemebene hängen, ohne eine signifikante Verbesserung des Gesamtsystems zu erreichen. Eine Verbesserung und weitere Steigerung der Anzahl der untersuchten Lösungsvarianten wird bei der mehrstufig-optimierenden Suche erreicht. Hier werden mehrere Varianten über mehrere Entwicklungsschritte untersucht. Erst wenn eine Variante als nicht funktionstüchtig eingestuft wird, wird deren Untersuchung terminiert. Es entsteht ein vollständiger Variantenbaum, der unter Umständen einen extrem hohen Grad an Komplexität erreichen kann.

Die linearen Methoden können leicht in den Entwurfszweig des V-Modells integriert werden.

Die zyklische Suche sieht die Möglichkeit des Rückgriffs auf eine der vorherigen Entscheidungsstufen vor. Eine Variante davon stellt das Spiralmodell dar.

Der Großteil der vorgestellten Strategien ist rein theoretisch und nur für ideale Anwendungen geeignet. In der Praxis liefert meistens eine Kombination der Ansätze das gewünschte Ergebnis. Um einen Designvorschlag für ein komplexes System, wie einen Sternsensor, ausarbeiten zu können, der, möglichst optimal, die gestellten Anforderungen und Randbedingungen erfüllt, empfiehlt sich eine **zyklische, mehrstufige und modellbasierte Designstrategie** mit ausgeprägtem Variantenmanagement.

5.2 Konzept des automatisierten Systementwurfs

Wie bereits in Kapitel 2 beschrieben wurde, ist die Kernaufgabe eines Systemingenieurs in der Sternsensorentwicklung, ein Multiparameter- und multidisziplinäres Optimierungsproblem zu lösen. Dabei sollen die Anforderungen und Randbedingungen der jeweiligen Anwendung berücksichtigt werden und das Ergebnis in Form der Systemspezifikation erfasst werden. Diese Aufgabe der Sternsensorentwicklung kann in drei Teilprobleme zerlegt werden:

- Variantenbildung,
- Bewertung der aktuellen Performance der Systemvarianten,
- Durchführung der Iterationen.

Als mögliches Lösungskonzept wurde der modellbasierte Einsatz in Kombination mit einem Optimierungsalgorithmus vorgeschlagen. Die Modellierung der Sternsensoren liefert die benötigten Informationen für die Bewertung der Varianten und der Optimierungsalgorithmus übernimmt die Variantenbildung und die Steuerung des zyklischen Ablaufs [BAL15].

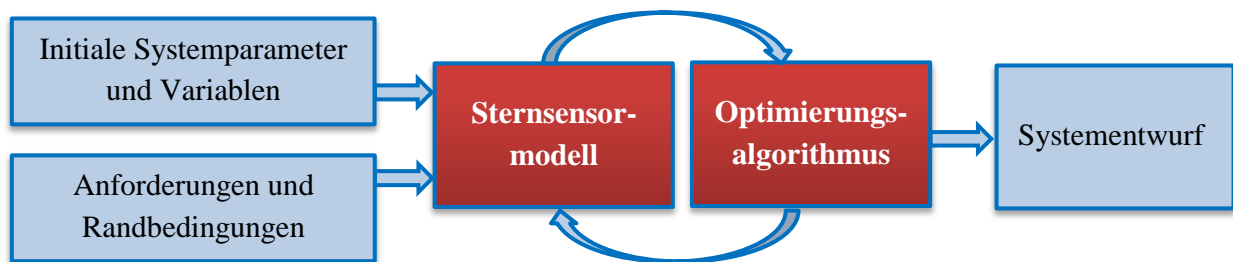


Abbildung 17. Konzept des Software-Optimierungstools

Das Modell und der Optimierungsalgorithmus bilden das Herzstück des Softwaretools zum Sternsensordesign, welches unter den vorgegebenen Randbedingungen und Optimierungszielen einen Entwurf eines Sternsensorsystems automatisch generieren soll (Abbildung 17).

Der vorgeschlagene Ansatz trägt zu einer Reduktion des Entwicklungsaufwandes in der Vordefinitionsphase B bei. Somit werden Zeit und Kosten gespart. Dies ist essenziell für Kurzzeit- und Budgetmissionen mit Pico- und Nanosatelliten. Die Aufgabe des Entwicklers besteht nun darin, das Systemmodell zu bilden, dann für eine Simulation die initialen Werte zu wählen und anschließend Anforderungen, Randbedingungen sowie Entwurfsziele zu formulieren. Das Tool führt eine automatische Entwurfsraumexploration durch und gibt als Ergebnis eine geeignete Konfiguration des Systems aus.

5.3 AROS – Forschungsvorhaben zur Verbesserung des Designablaufs von Sternsensoren

Um die Idee dieser neuen Designstrategie für Sternsensoren realisieren zu können und die technologische Kompetenz der Professur für Raumfahrttechnik an der Universität Würzburg im Bereich Sensorik weiter zu vertiefen, wurde im Jahr 2015 das Vorhaben „AROS - Entwicklung und Test neuartiger Software- und Hardware-Technologien für Miniatur-Sternsensoren für Pico- und Nanosatelliten“ (BMW, FKZ 50RM1522) initiiert. Das Vorhaben ist eng mit den theoretischen Überlegungen dieser Forschungsarbeit verzahnt [BAL14-1, FIS15] und wurde vom Autor diese Arbeit geleitet. Der Schwerpunkt des AROS-Vorhabens liegt jedoch mehr in der praktischen Anwendbarkeit des vorgeschlagenen Konzeptes. Im Rahmen des Vorhabens wurde die Funktionstüchtigkeit des Software-Tools zur Optimierung und Automatisierung des Designprozesses anhand von zwei Entwicklungsdurchläufen für beide obengenannten Satellitenklassen demonstriert [SCH15]. Als Ergebnis sind zwei Sternsensoren entstanden. Diese wurden gebaut und für den Einsatz im All qualifiziert [BAL15, BAL16].

5.4 Struktur des Optimierungstools

Die Struktur des Optimierungstool ist in der Abbildung 18 dargestellt. Der Schwerpunkt der Detaillierung wird an dieser Stelle zuerst auf das Modell des Sternsensormodells gelegt. Am Anfang der Modellbildung wird vorausgesetzt, dass eine Auswahl geeigneter und beschaffbarer COTS-Bildsensoren getroffen wurde.

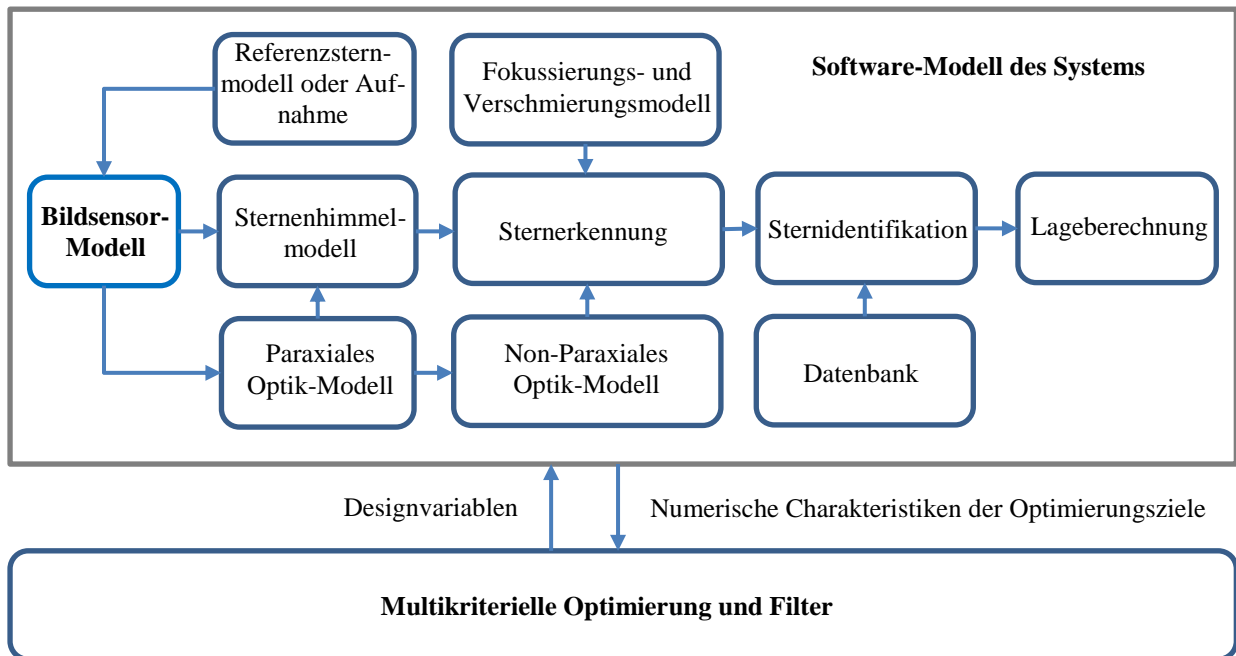


Abbildung 18. Struktur des Optimierungstools

Die vorgestellte Struktur des Sternsensormodells beinhaltet alle funktionalen Komponenten, die für eine erfolgreiche Lagebestimmung notwendig sind. Die Inhalte der Modellkomponenten sind ähnlich zu den Softwarefunktionen in Abbildung 6 (siehe Kapitel 2.1). Der Unterschied ist lediglich, dass die Bildaufnahme durch eine Simulation ersetzt wird. Ein multikriterieller Optimierungsalgorithmus liefert für jede Simulation einen Satz an Designvariablen. Anschließend bewertet er anhand einer Kostenfunktion die Ergebnisse der Modellierung der aktuellen Designvariante. Die Kostenfunktion beschreibt den Erreichungsgrad der Optimierungsziele. Führen die gewählten Designvariablen zu keiner Verbesserung des Gesamtsystems, so wird ein neuer Satz an Designvariablen generiert und mit der nächsten Iteration begonnen. Die Optimierung wird terminiert, wenn die Designziele erreicht sind oder eine gewünschte Anzahl an Iterationen durchgeführt wurde.

5.5 Modell des Sternsensormodells

Extrem wichtig bei der Modellbildung ist die Beschreibung der Zusammenhänge zwischen den einzelnen Modellkomponenten. Jede Komponente hat eine gewisse Anzahl an Inputs und Outputs. Input-Informationen kommen entweder von den anderen Komponenten oder von externen Quellen, wie z.B. Randbedingungen oder der gewünschten Systemspezifikation. Die Beschreibungen der Komponenten können einen unterschiedlichen Detaillierungsgrad erreichen und müssen dementsprechend aufeinander abgestimmt sein. Wenn das Verhalten einer

Modellkomponente durch Mangel an Information mathematisch nicht beschreibbar ist, kann diese durch verfügbare Messergebnisse aus einem Experiment ersetzt werden.

In den nachfolgenden Unterkapiteln werden einzelne Komponenten des Sternensormodells erläutert und gegebenenfalls mögliche Varianten der Modellbildung verglichen.

5.5.1 Abbildung der Sterne

Eine enorme Rolle bei der Auslegung eines Sternensensors spielt die Anzahl der sichtbaren Sterne im Sichtfeld. Die sichtbaren Sterne erscheinen dem Beobachter unterschiedlich hell. Die scheinbare Helligkeit hängt von der tatsächlichen Leuchtkraft eines Sterns und von der Entfernung zu diesem Stern ab.

5.5.1.1 Modell des Referenzsterns

Um die Leuchtkraft der Sterne außerhalb der Erdatmosphäre zu bestimmen, wird angenommen, dass Sterne ideale Strahler bzw. schwarze Körper sind. Die Intensität der Strahlung des schwarzen Körpers ist unabhängig von Größe und Form sowie chemischer Zusammensetzung des Objekts und hängt nur von seiner Temperatur ab. Das Planck'sche Strahlungsgesetz beschreibt die Intensität der Strahlung in Abhängigkeit von Temperatur und Wellenlänge [LIE02]:

$$I(T, \lambda) = \frac{2\pi \cdot h \cdot c^2}{\lambda^5} \cdot \frac{1}{e^{h \cdot c / \lambda \cdot k \cdot T} - 1} \quad (1)$$

mit

I – Intensität der Strahlung $\left[\frac{W}{m^2} \cdot m^{-1} \right]$,

π – Kreiszahl, $\pi = 3,14$,

h – Planck'sches Wirkungsquantum, $h = 6,626 \cdot 10^{-34} J \cdot s$,

c – Lichtgeschwindigkeit, $c = 2,997 \cdot 10^8 m/s$,

k – Boltzmann-Konstante, $k = 1,38 \cdot 10^{-23} J \cdot K^{-1}$,

λ – Wellenlänge in m ,

T – Temperatur des Schwarzen Körpers in K .

Durch die Integration der Intensitätsfunktion über das gesamte elektromagnetische Spektrum erhält man die Flussdichte (Bestrahlungsstärke) an der Oberfläche des schwarzen Körpers:

$$F' = \int_0^\infty I(T, \lambda) d\lambda = \sigma T^4. \quad (2)$$

Dieser Ausdruck ist bekannt als Stefan-Boltzmann-Gesetz und die Konstante $\sigma = 5,67 \cdot 10^{-8} W m^{-2} K^{-4}$ wird Stefan-Boltzmann-Konstante genannt.

Um das Modell eines Referenzsterns erstellen zu können, wird als Beispiel der nächste und bekannteste Stern, die Sonne, untersucht. Die Sonne hat eine Oberflächentemperatur von etwa

5800K. Dadurch ergibt sich eine Flussdichte auf der Oberfläche der Sonne, integriert über den Gesamtspektrum (Abbildung 19):

$$F' = \sigma T^4 = 5,67 \cdot 10^{-8} \cdot 5800^4 = 6,35 \cdot 10^7 W/m^2 = 63,37 MW/m^2 \quad (3)$$

Die Flussdichte der Strahlung sinkt proportional zum Quadrat des Abstands zur Quelle. Der mittlere Abstand zur Sonne beträgt $L = 149,65 \cdot 10^6 km$ und der Radius $R = 6,96 \cdot 10^5 km$. Wenn der Erdradius vernachlässigt wird, ergibt sich die sogenannte Solarkonstante:

$$F_{sonne} = F' \cdot \frac{R^2}{L^2} = F' \cdot k_{sonne} = 1370 W/m^2 \quad (4)$$

mit $k_{sonne} = \left(\frac{R}{L}\right)^2$ - Koeffizient der Proportionalität.

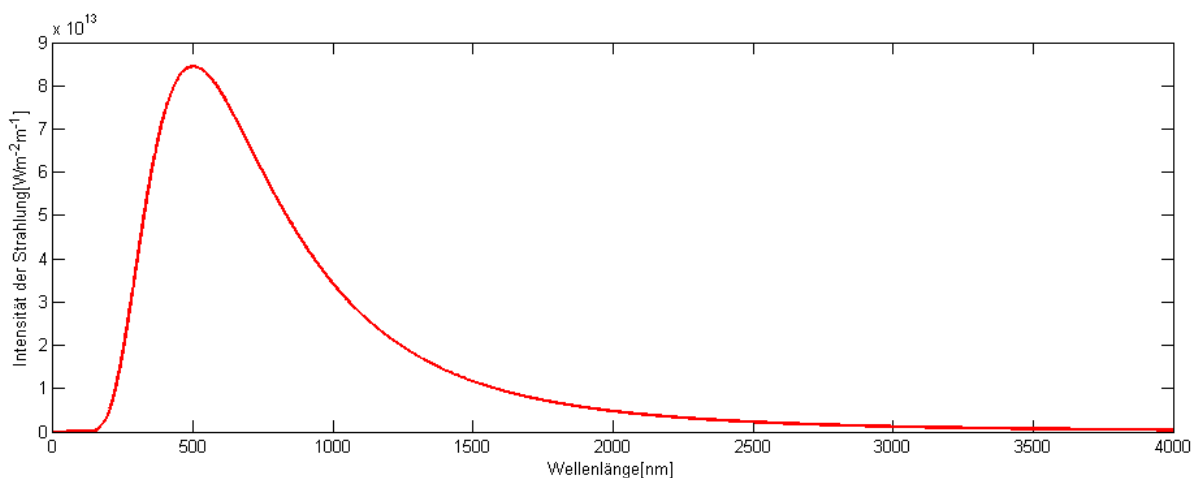


Abbildung 19. Sonnenspektrum als idealer schwarzer Körper

Die Eigenschaften des Referenzsterns werden als Bezugspunkt bei der Klassifikation der anderen Sterne benötigt. Als Referenzstern kann ein beliebiger Stern verwendet werden.

Die Inputs, der Output und der Zielbaustein für das Modell des Referenzsterns ist in Abbildung 20 dargestellt. Die Photonenflussdichte des Referenzsterns wird für die Modellierung der Belichtung des Bildsensors weiterverwendet. Der Wellenlängenbereich ist ein Parameter, der vom Nutzer spezifiziert wird. Die Magnitude wird durch den Stern, der simuliert werden soll, bestimmt.

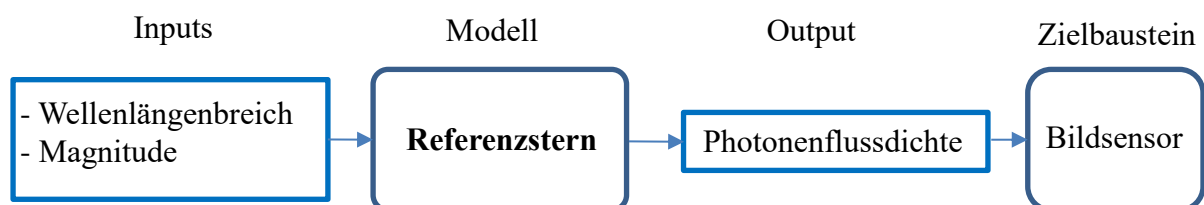


Abbildung 20. Environment des Referenzstern-Modells.

5.5.1.2 Klassifikation der Sterne

Im astronomischen Bereich existieren diverse Klassifizierungsmethoden für Sterne. Interessant für die weiteren Berechnungen ist die Spektralklassifizierung nach Harvard [KAR90].

Diese stellt einen Zusammenhang zwischen der Farbe bzw. der chemischen Zusammensetzung der Sternoberfläche und der Oberflächentemperatur auf. Die Haupttypen der Sterne werden durch große Buchstaben repräsentiert und werden in Unterklassen von 0 bis 9 unterteilt.

Tabelle 3. Spektralklassifikation der Sterne [HAN14, SCH18-1]

Klasse	Farbe	Oberflächentemperatur	Atome
O	Bläulich	30000-50000K	He II, He I, C III, O III, NIII
B	Bläulich-weiß	25000K	He I, HI, Ca II, Mg II, Si II
A	Weiß	10000K	H I, Ca II, Fe II, Si II, Mg II
F	Gelblich-weiß	7600K	H, K, Ca II, Fe I & II, Cr II, Ti II
G	Gelblich	6000K	Ca II, H, K, Metalle
K	Orange-gelblich	5100K	Metalle, TiO
M,C,S	Rötlich	3000K	TiO, Ca I, CN, CH, YO, LaO

Eine weitere und für die Sternidentifikation besonders wichtige Charakteristik ist die scheinbare Helligkeit der Sterne. Je nach Helligkeit werden die Sterne nach Größenklassen (Magnitude) differenziert. Die Klasse eines Sterns kann anhand der beobachteten Flussdichte berechnet werden. Als Referenz für dieses Klassifikationssystem wird die Flussdichte F_0 eines Sterns der Größenklasse 0 genommen. Traditionell dient der Stern Vega als visueller Nullpunkt. Die Magnitude eines Sterns mit der Flussdichte F_m wird folgendermaßen bestimmt [KAR90]:

$$m = -2,5 \lg \frac{F_m}{F_0}. \quad (5)$$

Daher können zwei beliebige Sterne der Magnituden m_1 und m_2 leicht verglichen werden. Für Ihre Flussdichten gilt:

$$m_1 - m_2 = -2,5 \lg \frac{F_1}{F_2} \quad \text{oder} \quad F_1/F_2 = 10^{-0,4(m_1-m_2)}. \quad (6)$$

Der gleiche Rechnungsweg kann nun relativ zu F_0 durchgeführt werden, um die Flussdichte und Strahlungsintensität eines Sterns anderer Magnitude zu bestimmen:

$$F_m = F_0 \cdot 10^{-0,4m} \quad \text{und} \quad F_m = k_m \int_0^\infty I_0(T_0, \lambda) d\lambda. \quad (7)$$

Einige Beispiele der scheinbaren Helligkeit für diverse Himmelskörper sind in Tabelle 4 zusammengefasst. Durch logarithmische Umkehrproportionalität haben schwächer leuchtende Objekte eine größere Magnitude-Klasse. Außerhalb der Stadt können Sterne mit einer Magnitude bis etwa +6 mit bloßen Augen erkannt werden. Innerhalb dieses Bereichs liegen auch die meisten Planeten.

Die scheinbare Helligkeit der Sterne hängt vom Spektralbereich des Lichts ab. In der Astronomie wird die scheinbare Helligkeit oft für den visuellen Bereich (V) um 550 nm angegeben. Es gibt auch weitere gebräuchliche Bereiche, wie Ultraviolett (U), Rot (R) etc. Diese werden für wissenschaftliche Beobachtungen mit optischen Teleskopen verwendet.

Die scheinbare Helligkeit im sichtbaren Wellenlängenbereich bildet nur ein Teil der Gesamtleistung ab. Die Sterne strahlen ja nicht nur in diesem Bereich, sondern auch im UV- und IR- sowie Röntgen-Spektrum. Das Maß der Gesamtstrahlung charakterisiert die scheinbare bolometrische Helligkeit:

$$m_{bol} = m_v - bc \quad (8)$$

mit dem bolometrischen Korrekturfaktor bc .

Die Sonne hat beispielweise eine scheinbare Magnitude m_v von $-26,75$ und eine bolometrische Helligkeit von $-26,82$ [GIE81]. Das entspricht eine Flussdichte außerhalb der Erdatmosphäre von 1370W/m^2 . Diese Größen sind unhandlich und es ist sinnvoll, zur Referenzmagnitude 0 überzugehen. Anhand von Formel (8) kann die Flussdichte für Sterne beliebiger Magnitude berechnet werden. Für die Magnitude 0 gilt:

$$F_0 = F_{sun} \cdot 10^{-0,4(0-m_{bol,sun})} = 1370 \cdot 10^{-0,4(0+26,82)} = 2,56 \cdot 10^{-8}\text{W/m}^2. \quad (9)$$

Um eine Verteilung der Strahlungsintensität über die verschiedenen Wellenlängen in der Nähe der Erdatmosphäre berechnen zu können, benötigt man die Größenklasse (Magnitude) und die Oberflächentemperatur.

Tabelle 4. Scheinbare Helligkeit einiger Himmelskörper [HAN14, SCH18-1].

Objekt	Scheinbare Helligkeit m_v [mag]
Sonne	-26,75
Vollmond	-12,7
Venus	-4,5
Mars	-2,91
Sirius	-1,46
Saturn	-0,47
Polarstern	+1,97
Für das menschliche Auge sichtbare Sterne	+6 (auf dem Land) +3 (in Städten)

Radiometrisches bzw. Fotometrisches Modell

Die Strahlungsenergie wird durch eine Glasbeschichtung oder den Einsatz eines dedizierten optischen Filters verringert bzw. absichtlich begrenzt, um die wellenlängenabhängige Aberrationen zu reduzieren. Typischerweise wird die Transmission auf den optischen Bereich von 380 nm bis 780 nm beschränkt. Manche Optiken sind auch im nahinfraroten Bereich durchlässig. Unter Berücksichtigung dieser Bedingung erhält man deutlich unterschiedliche Strahlungsintensitätskurven (siehe Abbildung 21) für Sterne der verschiedenen Spektralklassen und der Magnitude 0. Die Fläche unter der jeweiligen Kurve beschreibt die Strahlungsflussdichte des Sterns.

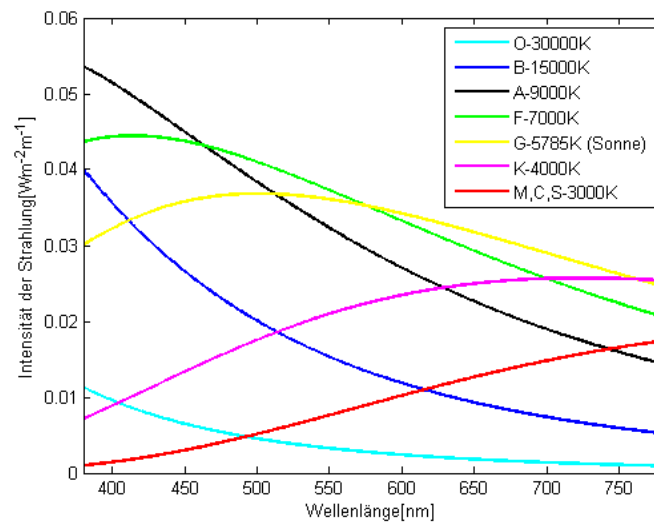


Abbildung 21. Strahlungsintensität der Sterne unterschiedlicher Klassen

Elektromagnetische Strahlung kann als ein Teilchenstrom (Photonenstrom) betrachtet werden. Die Photonen haben keine Ruhemasse aber besitzen Energie, die frequenzabhängig ist:

$$E = hf = \frac{hc}{\lambda} \quad (10)$$

Wenn man die Gleichung des Planck'schen Strahlungsgesetzes durch die Photonenenergie teilt, ist es möglich, die Flussdichte der Strahlung durch die Anzahl der Photonen diverser Wellenlänge darzustellen (siehe Abbildung 22).

Bei bekannter Öffnung der Optik kann die gesamte Anzahl der Photonen berechnet werden, die während der Belichtung von einer Sekunde die Pixel des Bildsensors treffen. Aber nicht jedes Lichtphoton, der die Bildsensoroberfläche erreicht, produziert in den Pixeln ein Photoelektron. Die Anzahl der produzierten Photoelektronen in der Sternabbildung ist von zwei Faktoren abhängig. Zuerst muss herausgefunden werden, wie gut das optische System einen Stern fokussieren und abbilden kann. In anderen Worten, wie groß die Fläche der Abbildung ist und wie die Energie zwischen den Pixel verteilt ist. Der zweite wichtige Faktor, der Anzahl der die generierten Photoelektronen beeinflusst, ist die Detektortechnologie, die durch Füllfaktor und Quanteneffizienz präsentiert wird.

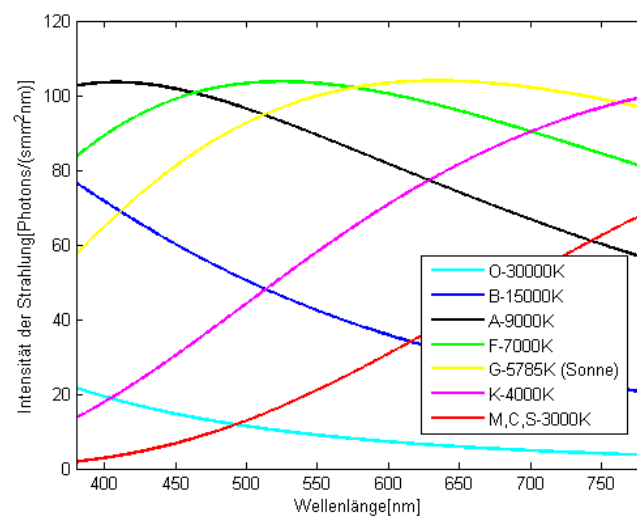


Abbildung 22. Photonenstromdichte der Sterne unterschiedlicher Klassen

5.5.2 Bildsensor

Derzeit sind zwei Typen von Flächendetektoren für digitale Bildaufnahmegeräte auf dem Markt verbreitet: CCDs (Charge-Coupled Devices) und APS (Active Pixel Sensors) mit CMOS-Technik. Diese beiden Technologien haben jeweils eigene Vorteile und Nachteile und werden als konkurrierend betrachtet [WÜL04, LIT01, LIT05, ERH08]. Die spezifischen Fachbegriffe aus dem Bereich Bildsensortechnik sind im Anhang A erläutert.

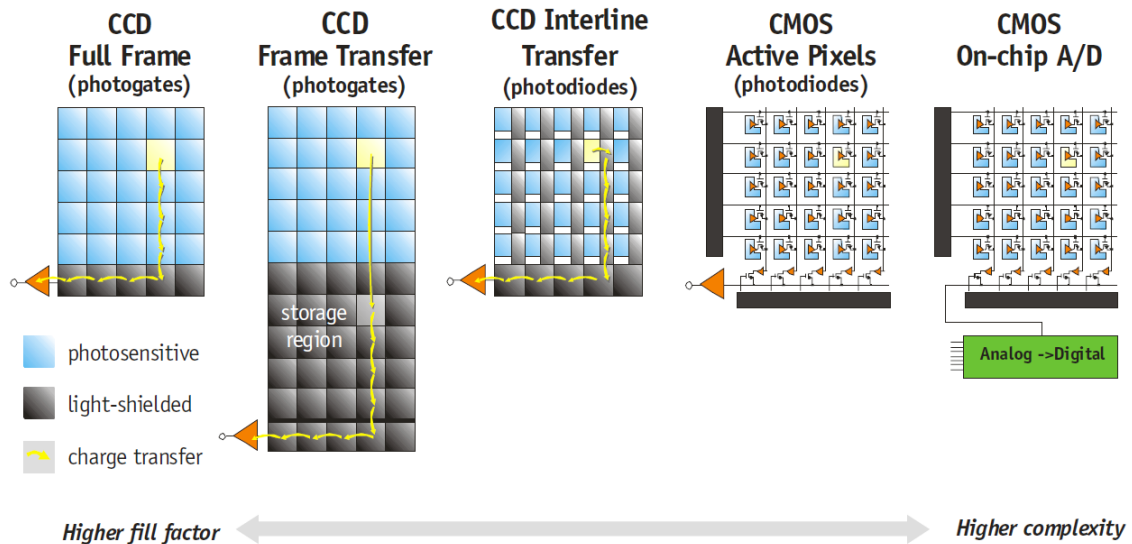


Abbildung 23. Architektur von CMOS und CCD Sensoren [DAL05]

5.5.2.1 CMOS Sensoren

In der einfachsten Realisierung hat jeder Pixel eines APS-CMOS-Sensors eine Photodiode und mindestens drei Transistoren. Der erste Transistor wird vor einer Belichtung angesteuert, um die Spannung an der Photodiode auf einen Anfangswert zu setzen (Reset). Während der Belichtung sinkt die Spannung in Sperrrichtung der Photodiode proportional zur Bestrahlungsstärke (Beleuchtungsstärke). Am Ende der Belichtung wird diese Spannung über den zweiten Transistor verstärkt und zur Verarbeitung mittels des dritten Schalttransistors zu einem bestimmten Auslesezeitpunkt weitergegeben.

Generell existieren zwei Typen von CMOS Sensoren: mit Rolling- oder Global-Shutter. Beim Rolling-Shutter werden Zeilen des Bildsensors zeitlich versetzt belichtet und ausgelesen. Dadurch können beweglichen Objekte im Bild verschmiert werden. Die Global-Shutter Sensoren nehmen dagegen das gesamte Bild zeitgleich.

Sensoren mit CMOS-Technologie haben im Vergleich zu CCD-Sensoren einen geringeren Energieverbrauch. Außerdem benötigen die CMOS-Sensoren weniger Peripherieelektronik, da einige Komponenten, wie z.B. AD-Wandler und Auslesesteuerung, bereits im Chip integriert sind. Die integrierte Ausleseelektronik der CMOS-Sensoren (siehe Abbildung 23) wandelt die durch Licht produzierten Photoelektronen direkt in eine Spannung um, dies schließt Effekte wie Blooming und Transfer-Smear fast komplett aus. Beim Auslesen der CMOS-Sensoren besteht die Möglichkeit, einen bestimmten Bildsensorbereich zu adressieren und abzufragen. Die Ausleseelektronik in den Pixeln absorbiert bzw. reflektiert jedoch einfallendes Licht und verringert somit den Anteil der lichtempfindlichen Fläche und als Konsequenz die Empfindlichkeit. Um die Lichtempfindlichkeit, zu erhöhen werden bei CMOS Sensoren

oft Mikrolinsen oder rückwärtige Belichtung (Back Side Illumination) verwendet. Beide Maßnahmen verändern die spektrale Verteilung der Quanteneffizienzkurve [JER10, CRO19]. Weiterhin gilt CMOS-Technologie allgemein als strahlungstoleranter.

5.5.2.2 CCD Sensoren

Auch CCDs haben gegenüber CMOS-Technik einige Vorteile. Sie sind lichtempfindlicher und haben einen größeren Füllfaktor von bis zu 100% bei den Vollbildsensoren, kombiniert mit einem mechanischen Verschluss (Shutter). Der größere Lichtempfindlichkeitsbereich trägt auch zu hoher Dynamik bei. Bei der Belichtung generiert das einfallende Licht in jedem Pixel des CCD Sensors eine Ladung, die proportional zur eingestrahnten Lichtstärke ist. Bevor eine Konvertierung der Ladung in eine Spannung erfolgt, werden diese in sogenannten Potenzialtöpfen gesammelt und ggf. in baugleichen, aber abgedeckten Pixeln gespeichert. Je nach Speichertechnologie unterscheidet man zwischen Interline-Transfer- und Frame-Transfer-Sensoren (Abbildung 23). Hier wird das Bild nach der Belichtung rasch in einen abgedeckten Bereich zwischen den Zeilen oder neben der lichtempfindlichen Fläche verschoben. Wenn der Transfer zu langsam abläuft, entstehen Smear-Effekte im Bild. Bei Überbelichtung eines Pixels kann die Ladung in die Nachbarpixel oder sogar in den abgedeckten Bereich übergehen. Es entstehen Blooming-Effekte. Da die Pixel von CCDs der Reihe nach mit nur einer Ausgangselektronik verstärkt und verarbeitet werden, sind Uniformität und Bildqualität sehr hoch. In CMOS-Sensoren hat jeder Pixel eine eigene Ausleseelektronik, deren Bauelemente produktionsbedingt nicht vollständig identisch sind. Dies verringert die Uniformität und führt zu einem erhöhten Rauschanteil in einem CMOS-Bild.

Mittlerweile haben CMOS-Sensoren CCDs fast komplett aus Automobil-, Industrie- und Consumer-Elektronikbereichen verdrängt. Nur für medizinische und naturwissenschaftliche Aufnahmen, wie z.B. in der Radiologie oder in der Astronomie, sind CCD-Sensoren immer noch im Einsatz. Ein detaillierter Vergleich der beiden Sensortypen ist in der nachfolgenden Tabelle zusammengefasst.

Aus der tabellarischen Analyse ergibt sich, dass CCD-Sensoren mit ihrer Bildqualität und Lichtempfindlichkeit immer noch sehr attraktiv sind. Diese Eigenschaften sind bei Aufnahmen in Dunkelheit besonders wichtig. CMOS-Sensoren können hingegen mit optimalen Eigenschaften in den Bereichen Größe- und Energieverbrauch, Strahlungstoleranz und allgemeine Handhabung überzeugen [FER19]. Eine wesentliche Rolle spielt die Beschaffbarkeit der Sensoren in kleinen Mengen. Hier punktet eindeutig die CMOS Technologie. Aus diesen Gründen ist sie für den Einsatz in Sternsensoren für Pico- und Nanosatelliten besser geeignet.

Tabelle 5. Vergleich der CMOS und CCD Sensortechnik

Kriterien	CCD-Sensor	CMOS-Sensor
Sensitivität, QE	++	+
Ausleserauschen	++	-
Auflösung	++	+
Geschwindigkeit	-	++
Dynamik	++	+
Blooming	--	++
Smear	--	++
Elektronischer Verschluss	-	++
Pixelkapazität	+	+
Pixelfehler	+	-
Optische Qualität, Füllfaktor	+	-
Pixeluniformität	++	-
Auslesen eines Fensters	-	++
On-Chip-Signalverarbeitung	--	++
Energieverbrauch	--	++
Preis	-	++
Größe	-	+
Strahlungstoleranz	-	+
Handhabung	-	+
Verfügbarkeit auf dem Markt	-	+

5.5.2.3 Bildsensor-Modell

Der Bildsensor dient als Schnittstelle zwischen Optik und Elektronik. Er wandelt elektromagnetische Strahlung in ein elektrisches Signal um. Nach der Auswahl des Bildsensors steht eine Reihe von Charakteristiken fest. Sie können nicht verändert werden und fließen in das Sensor-Modell als konstante Inputs bzw. Parameter ein. Diese sind in Abbildung 24 mit gepunkteten Rahmen markiert. Die Designvariablen, wie z.B. Belichtungszeit, sind im Laufe der Simulation veränderbar. Das Modell selbst besteht aus drei Teilen: einem photometrischen, einem geometrischen und einem elektrischen Teil. Diese können unabhängig voneinander betrachtet werden und geben separate Ergebnisse für die nachfolgenden Modellbausteine aus.

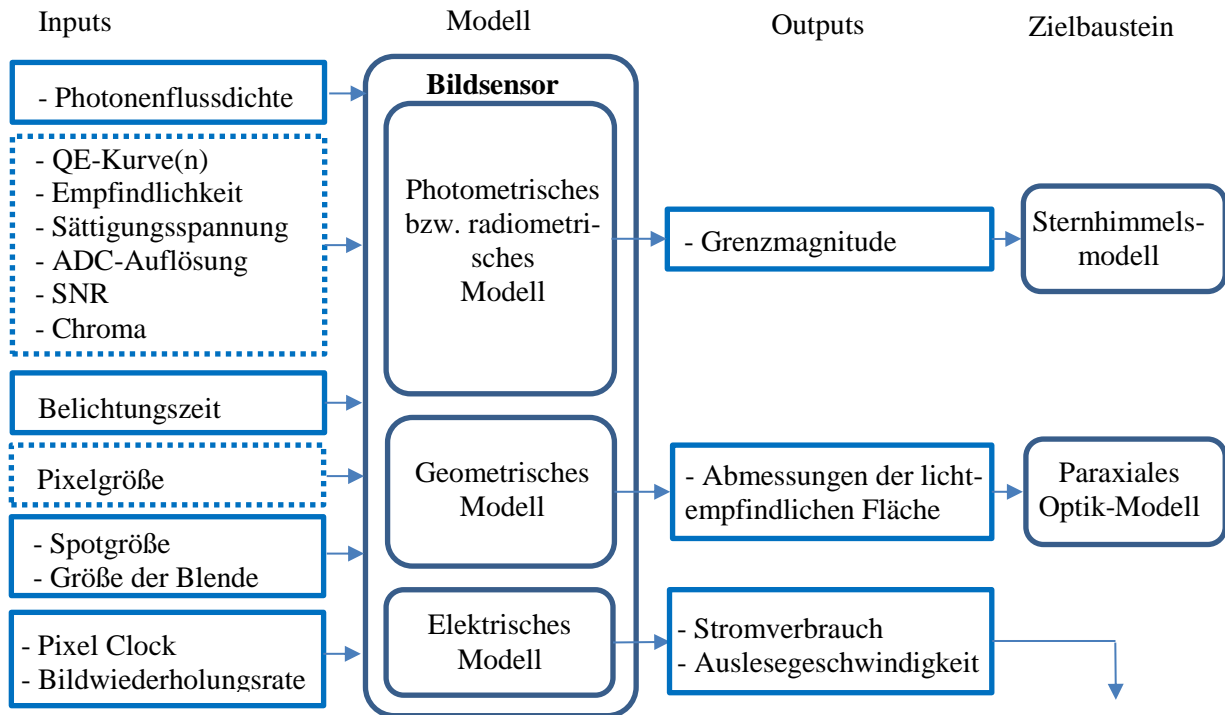


Abbildung 24. Bildsensor-Modell

- Photometrisches Modell

Anhand des photometrischen Modells kann die Grenzmagnitude der sichtbaren Sterne des Kamerasystems berechnet werden [LIE02]. Die Photonendichte, Belichtungszeit, Größe der Blende und Spotgröße liefern ausreichend Informationen für die Kalkulation der Anzahl an Photonen, die die lichtempfindliche Sensoroberfläche erreichen. Für die weitere Berechnung werden die opto-elektronischen Eigenschaften des Bildsensors benötigt. In der Dokumentation des Herstellers werden häufig nur typische Werte und Grenzwerte zur Quanteneffizienz, Empfindlichkeit und Dynamik der Sensoren angegeben. Es wird jedoch nicht erwähnt, unter welchen Bedingungen diese Charakteristiken ermittelt wurden. Je nach Hersteller können Testprozeduren, Testaufbau sowie Referenzlichtquellen recht unterschiedlich sein. Außerdem sind die Angaben in den Datenblättern häufig aus strategischen Gründen des Marketings übertrieben gut dargestellt, damit das Produkt attraktiver erscheint und sich von den Konkurrenzprodukten abhebt. Als Konsequenz kann die photometrische Modellierung alleine durch die Datenblätter der Hersteller mit großen Fehlern behaftet sein. Es ist deswegen für die vorliegende Arbeit besser, mit einem stark reduzierten Modell und realen Messwerten zu arbeiten.

Eine andere Methode die Grenzmagnitude bestimmen zu können, ist die Aufnahme des realen Sternhimmels unter bekannten Bedingungen. Dabei müssen die Öffnung des optischen Systems und die Belichtungszeit protokolliert werden. Für die Aufnahme können industrielle Kameras oder Demokits des Detektorherstellers mit qualitativer Industrieoptik verwendet werden.

Ein Beispiel für eine solche Vorgehensweise sind Feldversuche mit einem Sensordemokit für den monochromen „Jade“-Sensor EV76C454BBT-EQV vom Hersteller E2V. Hierfür wurde

ein lichtstarkes CCTV-Objektiv von Pentax mit einer Brennweite von 25 mm und einer Blendenzahl von 1,6 ($\text{Ø}15,625 \text{ mm}$) verwendet (Abbildung 25). Eine Besonderheit des Objektivs ist, dass es einen Teil des IR-Spektrums durchlässt. Das Objektiv hat einen standardisierten S-Mount-Anschluss und konnte über einen C-Mount-Adapter leicht an das Demokit angebunden werden.



Abbildung 25. S-Mount-Objektiv 25mm/1.6

Die Aufnahmen wurden im Winter bei Minustemperaturen und klarem Sternhimmel in der Nähe von Würzburg gemacht. In Abbildung 26 ist eine Aufnahme des Sternbildes „Größer Bär“ dargestellt. Bei dieser Testaufnahme wurde eine Belichtungszeit von 100 ms und eine maximale Verstärkung von 4 eingestellt. Die Bittiefe der Aufnahme beträgt 8 Bit. Das entspricht bei maximaler Helligkeit einem Wert von 255.

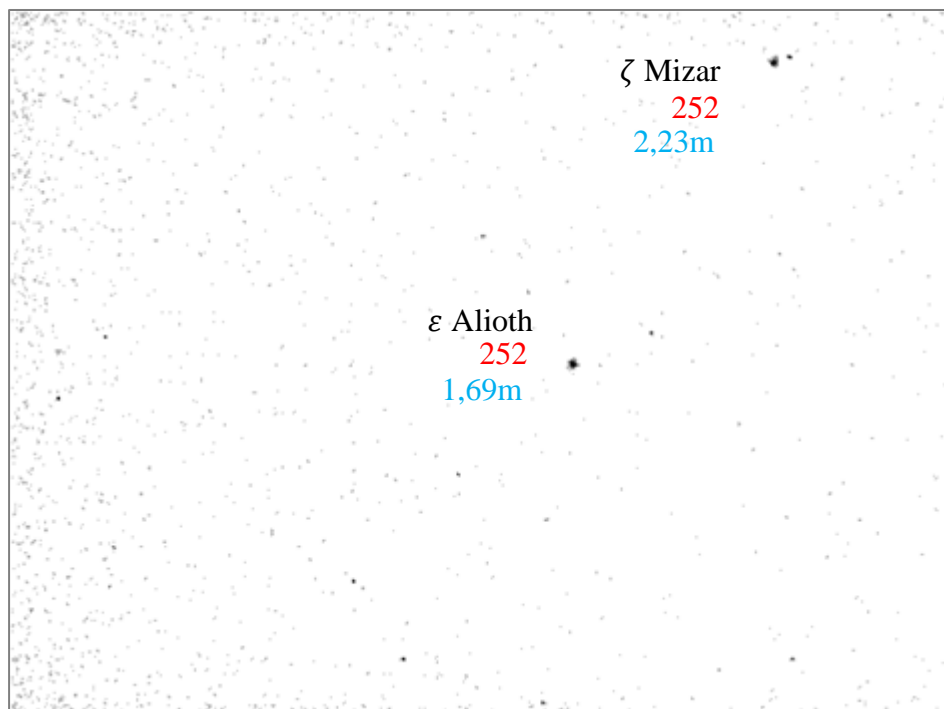


Abbildung 26. Invertierte Testaufnahmen des Sternenhimmels

Die Analysen zeigen, dass Sterne bis Magnitude 6,7 auf mehrere Pixel abgebildet werden und deutlich sichtbar sind. Sogar Sterne bis zu Magnitude 7 konnten vom Hintergrund unterschieden werden. Diese sind jedoch nur auf ein bis zwei Pixeln sichtbar und bieten keine sichere Referenz für nachfolgende Sternidentifikation. Um die Ergebnisse zu verdeutlichen, wurde das Bild mit einem Schwellenwertfilter von 27 bearbeitet und die hellen Pixel markiert. Abbildung 27 zeigt das Ergebnis der Bearbeitung. Die identifizierten Sterne wurden zusätzlich

mit ihrer Magnitude und mit maximaler Pixelhelligkeiten versehen. Die Kreise im Bild markieren Pixelfehler bzw. Störungen, z.B. von vorbeifliegenden Flugzeugen.

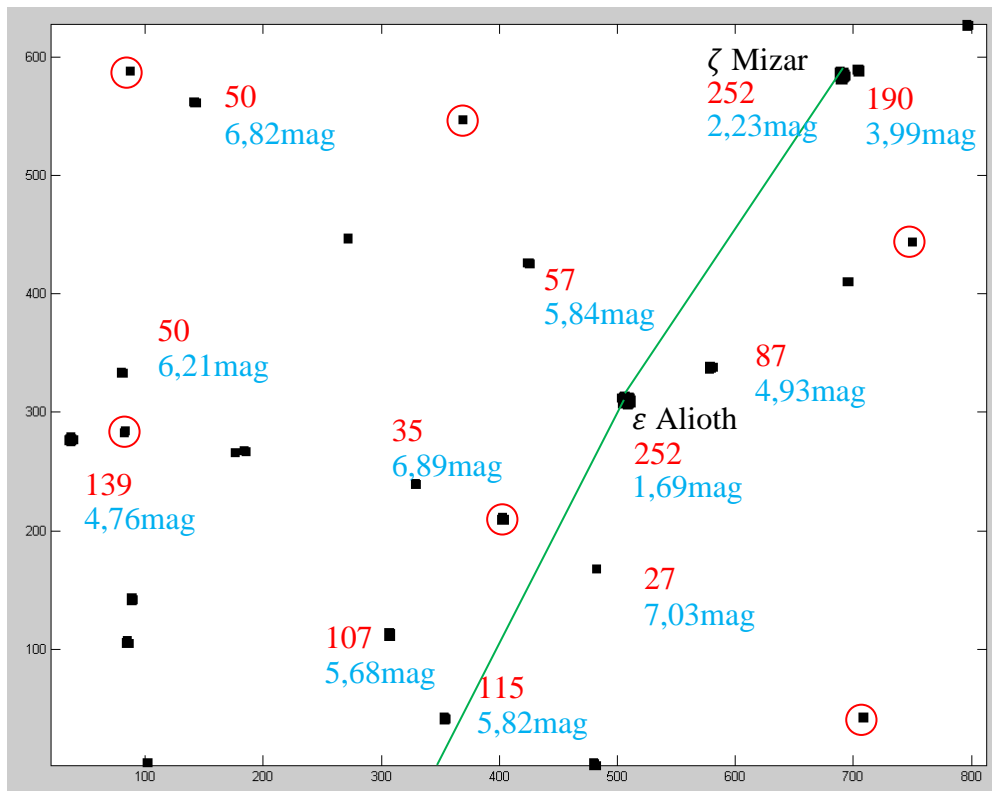


Abbildung 27. Bearbeitete Testaufnahme

Wenn eine oder mehrere Referenzaufnahmen gemacht wurden, können die Ergebnisse für Systeme mit anderer Optik oder Belichtungszeit umgerechnet werden:

$$m = m_{ref} + 2,5 \lg \left(\frac{F_{ref}}{F} \right) = m_{ref} + 2,5 \lg \left(\frac{d^2}{d_{ref}^2} \cdot \frac{t_{ref}}{t} \right) \quad (11)$$

mit F_{ref} , F - Flussdichte,

d_{ref} , d - Durchmesser der Eintrittspupillen der Referenzoptik und der zu berechnenden Optik,

t_{ref} , t - Belichtungszeiten für Referenz- und Zielsystem.

Diese Formel kann nur unter der Annahme verwendet werden, dass die Abbildungen des betrachteten Sterns von den beiden Optiken gleichgroß sind. Ansonsten muss die obengenannte Formel mit dem Verhältnis der Sternflächen in den Abbildungen vervollständigt werden.

Außerdem berücksichtigt die Formel den Einfluss der Atmosphäre nicht. Die Absorption bzw. Transmission der Atmosphäre ist wellenlängenabhängig. Im sichtbaren Bereich wird etwa 20% der extraterrestrischen Sonnenstrahlung absorbiert. Aus diesem Grund ist die Grenzmagnitude der sichtbaren Sterne außerhalb der Atmosphäre größer. Dieser Effekt soll bei der Berechnung des On-Board-Sternkatalog berücksichtigt werden oder kann durch Verringerung der Belichtungszeit kompensiert werden.

- Geometrisches Modell

Das geometrische Modell ist relativ einfach. Als Ergebnis soll nur die Größe der lichtempfindlichen Fläche berechnet werden:

$$\begin{aligned} V &= l_{v_pix} N_{ver}, \\ H &= l_{h_pix} N_{hor}, \end{aligned} \tag{12}$$

mit

l_{v_pix}, l_{h_pix} - Vertikale und horizontale Pixelgröße,

N_{ver}, N_{hor} - Vertikale und horizontale Anzahl an Pixeln der aktiven Sensorfläche.

Manchmal können die Werte für V, H direkt aus dem Datenblatt entnommen werden. Man muss jedoch beachten, dass N_{ver} und N_{hor} auch inaktive Pixel beinhalten können, wenn der Interlacing- oder Subsampling-Mode des Bildsensors verwendet wird.

- Elektrisches Modell

Bei der Modellierung des elektrischen Systems geht es um eine Abschätzung des Stromverbrauches. Es wird davon ausgegangen, dass der Bildsensor eine konstante Anzahl an Ausgangspins treibt und von einer externen Quelle mit einer bestimmten gleichbleibenden Grundfrequenz getaktet wird.

Folgende Einstellungen des Detektors können den Stromverbrauch beeinflussen:

- Interne Taktfrequenz, gegeben durch Einstellungen des Frequenzteilers,
- Belichtungszeit, t_{exp}
- Bildausgaberate, fps oder t_{frame}

Alle drei Größen haben untereinander wechselseitige Wirkungen. Die interne Taktfrequenz beeinflusst Belichtungsdauer und Auslesegeschwindigkeit der Daten (t_{read_out}) und somit auch die Bildausgaberate. Längere Belichtungszeiten benötigen eine langsamere Taktfrequenz und umgekehrt. Wenn eine höhere Bildwiederholungsrate benötigt wird, muss die Belichtungszeit verkürzt und die Taktrate erhöht werden.

$$\begin{aligned} t_{frame} &= t_{exp} + t_{read_out} \\ t_{frame} &= \frac{1}{fps} \end{aligned} \tag{13}$$

Bei einer Sternbeobachtung und bei astronomischen Anwendungen dominiert die Belichtungszeit, die Auslesegeschwindigkeit der Daten ist vernachlässigbar, $t_{exp} \gg t_{read_out}$ und

$$fps = \frac{1}{t_{exp}}. \tag{14}$$

Nun wird ein serieller, getriggertes Modus eines CMOS-Sensors betrachtet. In diesem Modus, laufen die Belichtung und das Sensorauslesen nach einem Triggersignal nacheinander ohne Überlappung ab. Abbildung 28 zeigt eine Stufenapproximation des Stromverbrauchs. Diese ist näherungsweise für Global- und Rolling-Shutter-Sensoren gültig.

Ein Modus mit Überlappung des Belichtungs- und Ausleseintervalls ist nur bei Sensoren mit Global-Shutter und 5T-Technologie möglich [FER16]. In diesem Modus laufen die Belichtung des neuen und das Auslesen des vorherigen Bildes parallel ab.

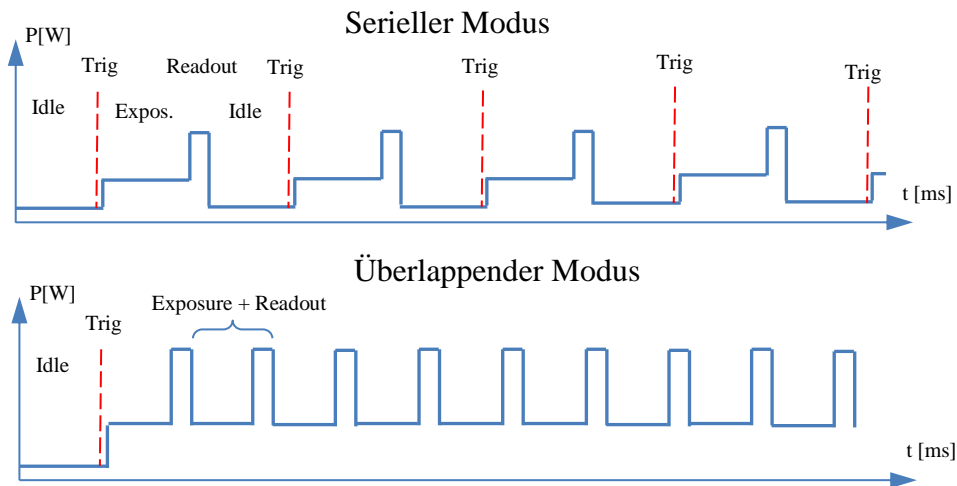


Abbildung 28. Verschiedene Betriebsmodi eines CMOS-Sensors

Wie zu erkennen ist, ist der durchschnittliche Stromverbrauch proportional zur Bildfrequenz, fps . In Datenblättern der Sensoren ist der Stromverbrauch häufig nur für einen bestimmten fps -Wert eingegeben. Oft entsprechen diese Werte der maximal möglichen Bildwiederholungsrate. Die restlichen Punkte der Kurve zur Stromaufnahme wie in Abbildung 29 müssen experimentell bestimmt werden.

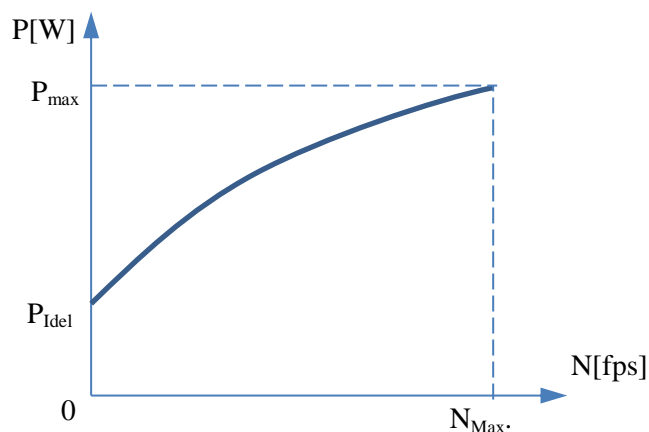


Abbildung 29. Typischer Stromverbrauch eines Bildsensors

Zusammengefasst lässt sich sagen, dass durch die mangelhaften bzw. nicht präzisen Angaben in Datenblättern und Entwicklungshandbüchern eine vollständige mathematische Beschreibung eines Bildsensors nicht möglich ist. In solchen Fällen ist die Anschaffung eines Demo-

kits oder -geräts mit dem zu untersuchenden Bildsensor empfehlungswert und sinnvoll. Anhand des Demogeräts können nicht nur die Referenzaufnahmen, sondern auch Abschätzungen zur Energieaufnahme unter den gewünschten Arbeitsbedingungen angefertigt werden. Außerdem können so Informationen über die Handhabung und andere Besonderheiten des Sensors gewonnen werden.

5.5.3 Paraxiale Optik

Die Qualität eines optoelektronischen Messsystems zeichnet sich nicht durch einen hochwertigen Bildsensor oder eine hervorragende Optik, sondern durch ein ausgewogenes Zusammenspiel beider Komponenten aus. Die primäre Aufgabe der Optik ist es, die gewünschten Objekte möglichst fehlerfrei auf die Detektorgröße zu skalieren. Die Modellierung eines optischen Systems beginnt mit einer linearen Approximation der Strahlengänge, der sogenannten paraxialen Optik. Im gewählten System aus optischen Flächen, Linsen oder Spiegeln wird die Propagation der Strahlen rein geometrisch berechnet. Als Ergebnis liefert das lineare Modell die wichtigen Informationen, wie Optiköffnung und Sichtfeldgröße. Diese werden für eine weitere Berechnung der Anzahl der sichtbaren Sterne verwendet (siehe Abbildung 30).

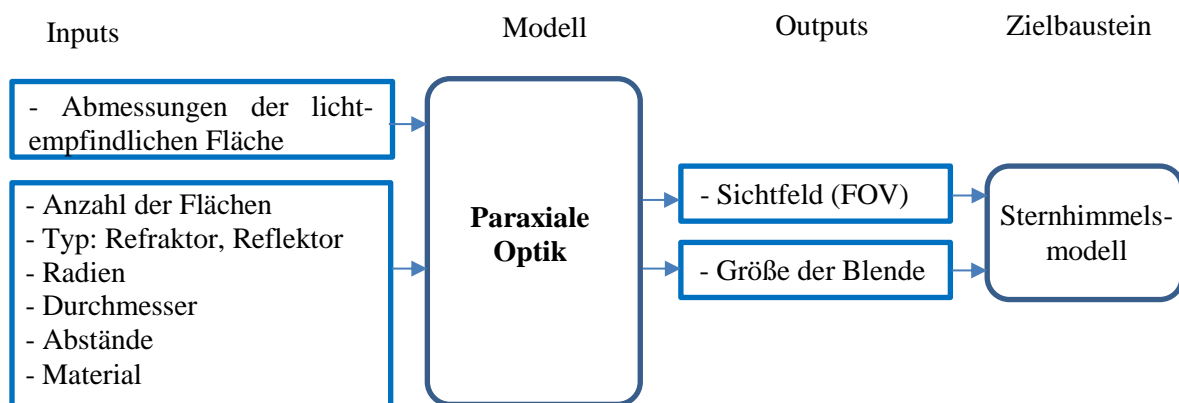


Abbildung 30. Paraxiales Modell der Optik

5.5.3.1 Geometrische Optik

In der geometrischen Optik gelten drei Gesetze, die die Ausbreitung von Lichtstrahlen beschreiben. Diese drei Gesetze werden für die Berechnung eines Strahlenganges benötigt [OSH85, SCH87]:

Das **Transmissionsgesetz** besagt, dass sich die Lichtstrahlen in einem homogenen Medium linear ausbreiten.

Das **Reflexionsgesetz** besagt, dass der Einfallswinkel u_i genau so groß wie der Reflexionswinkel u_r ist und beide Winkel zusammen mit dem Lot zur reflektierenden Oberfläche in einer Ebene liegen (siehe Abbildung 31):

$$u_i = u_r. \quad (15)$$

Das **Brechungsgesetz** beschreibt den Verlauf eines Strahls durch eine Grenzfläche zwischen zwei Medien mit unterschiedlichen Ausbreitungsgeschwindigkeiten des Lichts:

$$\frac{\sin u_i}{\sin u_j} = \frac{c_i}{c_j} \quad (16)$$

mit

u_i - Einfallswinkel im Medium i ,

u_j - gebrochener Winkel im Medium j ,

c_i, c_j – Ausbreitungsgeschwindigkeiten des Lichts in Medien i und j .

Das Operieren mit solchen großen Werten wie Lichtgeschwindigkeiten c_i und c_j ist nicht praktisch. Deshalb nutzt man eine relative Größe, die sogenannte Brechzahl des Mediums i :

$$n_{i,abs} = \frac{c_{vakuum}}{c_i} \text{ oder } n_i = \frac{c_{luft}}{c_i} \quad (17)$$

Unter Berücksichtigung dieser Gleichungen erhält das Brechungsgesetz die Form:

$$n_i \sin u_i = n_j \sin u_j \quad (18)$$

Die Brechzahl eines Mediums ist nicht konstant und ist eine Funktion der Wellenlänge λ . Im sichtbaren Bereich des Spektrums nimmt die Brechzahl von Violett zu Rot ab. Die Abhängigkeit $n(\lambda)$ wird als Dispersion des Mediums bezeichnet. Im Schott- oder Ohara-Glaskatalog wird die Dispersion für jede Glassorte über eine spezielle nicht-lineare Dispersionsfunktion beschrieben. Diese Funktion erlaubt eine sehr genaue Bestimmung der Brechzahl für verschiedene Lichtwellenlängen. Die Nutzung der Dispersionsfunktion ist für einfache Rechnungen wie beim Paraxial-Modell nicht besonders angemessen. Deshalb wird hier mit nur einer Wellenlänge üblicherweise, um 550 nm, agiert.

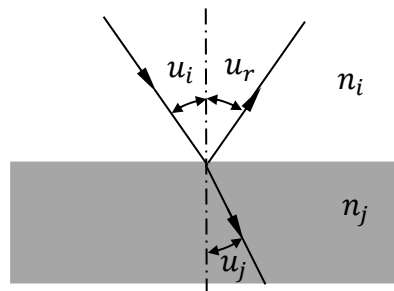


Abbildung 31. Reflexion und Brechung der Strahlen

Falls doch verschiedene Lichtwellen analysiert werden sollen, hat sich eine andere Darstellungsform der Brechung und der Dispersion anhand der Hauptbrechzahl n_e und der Abbeschen Zahl v_e bewährt. Die Hauptbrechzahl beschreibt die Brechung des gelbgrünen Lichtes mit der Wellenlänge $\lambda = 546,1$ nm an der sogenannten Spektrallinie e (siehe Anhang B). Die Abbesche Zahl ist eine relative Größe:

$$v_e = \frac{n_e - 1}{n_{F'} - n_{C'}} \quad (19)$$

Die Brechzahldifferenz $n_F - n_C$ heißt Hauptdispersion. Eine hohe Abbesche Zahl bedeutet eine niedrige Dispersion des Glases. Für bestimmte Glassorten sind die Hauptbrechzahl n_e und die Abbesche Zahl im Lageplan der optischen Gläser (siehe Anhang C) dargestellt. Anstelle der Hauptbrechzahl n_e und der Abbeschen Zahl v_e werden auch häufig die Größen n_d und v_d verwendet. Diese werden entsprechend für die Spektrallinie d und die Teildispersion $n_F - n_C$ berechnet.

5.5.3.2 Strahlenverlauf im Paraxialgebiet

In einer vereinfachten Beschreibung eines Strahlengangs durch ein optisches System wird angenommen, dass der Neigungswinkel des Strahls relativ zur optischen Achse sehr klein ist. Mit dieser Annahme reduziert sich die Taylor-Entwicklung für trigonometrische Funktionen auf die erste Komponente:

$$\begin{aligned}\sin u &= u - \frac{u^3}{3!} + \frac{u^5}{5!} - \dots \approx u, \\ \cos u &= 1 - \frac{u^2}{2!} + \frac{u^4}{4!} - \dots \approx 1, \\ \tan u &\approx u.\end{aligned}\tag{20}$$

Somit lässt sich das Brechungsgesetz linearisieren und geht dann in folgende Form über:

$$n_i u_i = n_j u_j.\tag{21}$$

Die paraxiale Linearisierung des Brechungsgesetzes erlaubt mit wenig Berechnungsaufwand, ein System aus mehreren optischen Flächen zu modellieren. Die wichtigsten Parameter, wie effektive Brennweite, Lage und Durchmesser der Aperturblende, der Eintritts- und Austrittspupillen sowie der Feldblenden, können so ermittelt werden. Außerdem kann eine erste Schätzung der Farbfehler (chromatische Aberrationen) durchgeführt werden.

Abbildung 32 zeigt den Verlauf eines Strahls durch zwei Flächen k und $k+1$. Als Fläche bezeichnet man eine Grenze zwischen zwei Medien. Eine Linse hat zum Beispiel zwei sphärische bzw. asphärischen Flächen. Ein Spiegel hat dagegen nur eine reflektierende Fläche.

In jedem Berechnungsschritt des Strahlenganges werden zwei Abläufe simuliert: Übergang von einer Fläche zur anderen, wobei der Strahl immer im gleichen Medium bleibt, und Flächendurchgang bzw. Reflektion, dabei interagiert der Strahl mit einem anderen Medium. Diese zwei Abläufe werden entweder durch die Matrizenmethode [PED08] oder durch zwei einzelne Gleichungen nach [OSH85] beschrieben. Das letzte Verfahren wird im weiteren Verlauf des Kapitels vorgestellt.

Der Übergang zwischen Punkt A_k und A_{k+1} wird durch die Übergangsgleichung dargestellt. Ziel der Berechnung ist es die neue Höhe y_{k+1} des Durchgangspunktes A_{k+1} zu bestimmen:

$$y_{k+1} = y_k + u_k t_k.\tag{22}$$

In der Übergangsgleichung wurde angenommen, dass Abstand t_k zwischen zwei Scheitelpunkten der optischen Oberflächen für einen kleinen Winkel u_k gleich dem Abstand $B_k B_{k+1}$ ist.

$$t_k \approx B_k B_{k+1}. \quad (23)$$

Wenn der Strahl dann auf die Fläche k+1 trifft, wird er entweder gebrochen oder reflektiert. Dies wird durch die Gleichung zum Flächendurchgang beschrieben:

$$n_{k+1} u_{k+1} = n_k u_k - \frac{n_{k+1} - n_k}{R_{k+1}} y_{k+1}. \quad (24)$$

Zur Einfachheit werden folgende Variablen eingefügt: $nu_{k+1} \equiv n_{k+1} u_{k+1}$, $nu_k \equiv n_k u_k$, und die Flächenkrümmung $c_{k+1} = 1/R_{k+1}$. Die **Durchgangsgleichung** nimmt dadurch folgende Form an:

$$nu_{k+1} = nu_k - y_{k+1}(n_{k+1} - n_k)c_{k+1}. \quad (25)$$

Die **Übergangsgleichung** kann auch in Abhängigkeit von nu umgeformt werden. Diese Schreibweise ist praktischer für die Berechnung am Computer:

$$y_{k+1} = y_k + \frac{nu_k \cdot t_k}{n_k}. \quad (26)$$

Die Gleichungen (25) und (26) beschreiben den Strahlengang des Lichtes durch ein optisches System aus einer beliebigen Anzahl optischer Flächen. Dabei müssen die Vorzeichen der Gleichungskomponenten streng beachtet werden. Die Vorzeichenregeln sind in Anhang D zusammengefasst. Wenn der Strahl auf einer optischen Fläche reflektiert wird, werden die Vorzeichen der Brechzahl n_k und des Abstandes t_k zwischen den Flächen invertiert.

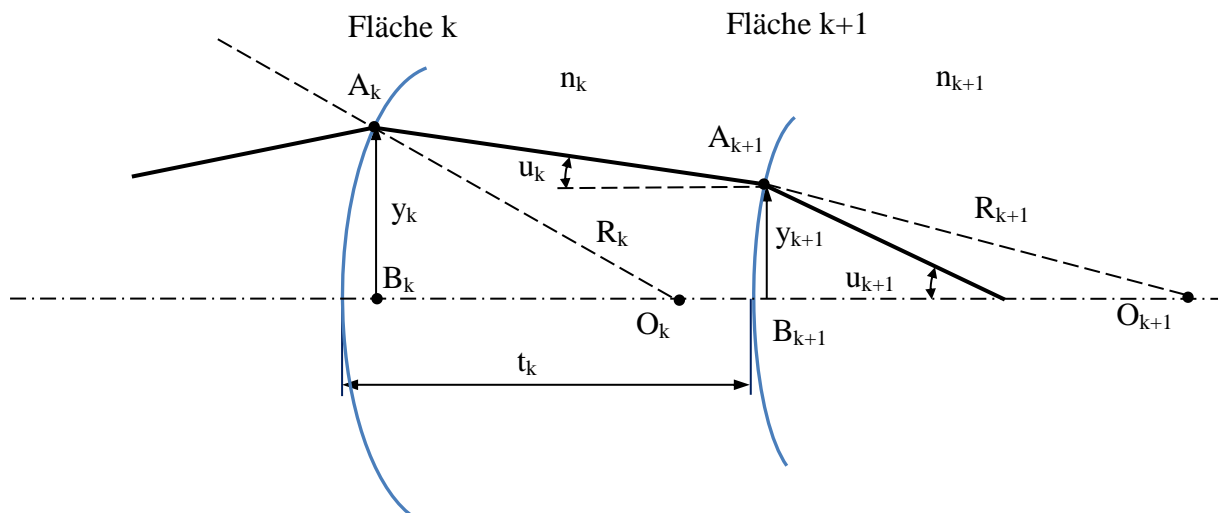


Abbildung 32. Paraxialer Strahlengang durch zwei Flächen [OSH85]

Ein optisches System, bestehend aus mehreren Linsen bzw. Flächen, ist in Abbildung 33 dargestellt. Alle j optischen Flächen bzw. Ebenen werden durchnummeriert, beginnend mit der Objektebene Nummer $i = 0$. Die erste optische Fläche erhält die Nummer 1 und die letzte die Nummer j . Die Abbildungsebene schließt die Kette mit der Nummer $j + 1$ ab. Die Brechzahlen zwischen den Flächen und Ebenen bekommen die Bezeichnungen n_0 bis n_j . Die Ergebnisse der Berechnung werden häufig in einer tabellarischen Form dargestellt.

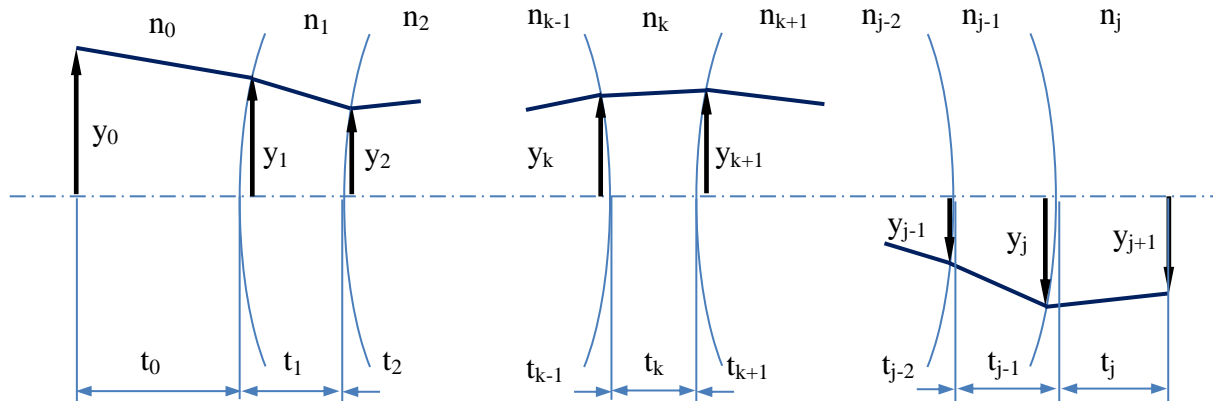


Abbildung 33. Strahlengang durch mehrere optische Flächen [OSH85]

Einer der wichtigsten Parameter des optischen Systems ist die **Aperturblende**. Die Aperturblende ist sozusagen die engste Stelle im System und grenzt den Strahlenbündeln ein, wodurch die Lichtmenge, die durch das Gesamtsystem geht, limitiert wird. Um diese engste Stelle in einem optischen System finden zu können, wird die Skalierbarkeit der Paraxialrechnung ausgenutzt. Die Höhe des Durchgangspunktes y_{i+1} ist nämlich proportional zum Winkel u_i . Das bedeutet, dass der axiale Anfangswinkel u_0 beliebig skaliert werden kann. Dabei erhält man eine ganze Familie an axialen Strahlen, die ein Bündel mit unterschiedlichen Anfangswinkeln an der optischen Achse bilden. Wenn ein Strahl bei Skalierung den Rand eines optischen Elementes mit dem Durchmesser $2r_i$ erreicht, bezeichnet man ihn als Randstrahl. Der Skalierungskoeffizient beträgt dann

$$\gamma_i = \left| \frac{r_i}{y_i} \right| \quad (27)$$

mit

r_i - Außenradius der Fläche i bzw. des optischen Elements

y_i - Höhe des axialen Strahles bei Fläche i .

Um die Lage der Aperturblende im System zu bestimmen, wird ein beliebiger axialer Strahlengang durch das System unter paraxialen Bedingungen berechnet. Die optische Fläche bzw. Ebene i mit kleinstem γ_i ist die engste Stelle des optischen Systems und somit die Aperturblende (aperture stop):

$$as = \min_i \left| \frac{r_i}{y_i} \right| \quad (28)$$

Die Aperturblende kann eine Linse bzw. Linsenfassung oder eine extra eingefügte mechanische Komponente, z.B. eine Scheibe, sein.

Es ist auch wichtig zu wissen, wie die Abbildung der Aperturblende aus der Perspektive des Objekts und dessen Abbildung erscheint. Man nennt diese Abbildungen **Eintrittspupille** und **Austrittspupille**. Die Eintrittspupille bestimmt, wie viel Strahlungsleistung die Optik aufnehmen kann. Die Austrittspupille zeigt, auf welche Fläche die aufgenommene Leistung in der Bildebene verteilt wird.

Die Größe der Eintritts- und Austrittspupillen kann anhand der optischen (Lagrange-) Invariante I bestimmt werden:

$$I = \bar{y}_k n_k u_k - y_k n_k \bar{u}_k \quad (29)$$

Die optische Invariante besagt, dass die vorgestellte Differenz für zwei unabhängige Strahlen, z.B. für einen axialen Strahl ($y_0 = 0, u_0 \neq 0$) und einen schrägen Strahl ($\bar{y}_0 \neq 0, \bar{u}_0 \neq 0$), beim Übergang k zu einem beliebigen Medium n konstant ist.

Es wird angenommen, dass ein Objekt mit der Höhe h als Abbildung mit der Höhe h' in der Bildebene dargestellt wird. Für die Berechnung werden zwei Strahlen $y_0 = 0$ und $\bar{y}_0 = h$ durch das optische System propagiert. Dies bedeutet, dass

$$I = h n u_0 - 0 \cdot \bar{n} \bar{u}_0 = h n_0 u_0 \quad (30)$$

und von der anderen Seite

$$I = h' n u_j - 0 \cdot \bar{n} \bar{u}_j = h' n_j u_j. \quad (31)$$

Somit ergibt sich

$$h n_0 u_0 = h' n_j u_j. \quad (32)$$

Als Abbildungsmaßstab M wird folgendes Verhältnis definiert:

$$M = \frac{h'}{h} = \frac{n_0 u_0}{n_j u_j} \quad (33)$$

Wenn die Brechzahlen der Medien vor und nach dem optischen System gleich sind, nämlich $n_0 = n_j$, ergibt sich:

$$M = \frac{u_0}{u_j} \quad (34)$$

Eintritts- und Austrittspupille sind Abbildungen der Aperturblende in Objekt- und Bildebene. Um die Größe und Lage dieser Pupillen zu finden, soll die Propagation eines Hauptstrahls ($\bar{y}_{as} = 0$) durch das System, beginnend an der Aperturblende, in Richtung der Objektebene und der Bildebene untersucht werden.

Bei der Abbildung der Aperturblende wird der Strahlendurchlauf über den linken objektseitigen Teil des optischen Systems rückwärts gerechnet. Nach Definition des Hauptstrahls schneidet dieser die optische Achse bei der Aperturblende. Somit ist $\bar{y}_0 = 0$.

$$\bar{y}_1 = \bar{y}_0 + \bar{u}_0 \bar{t}_0 = \bar{u}_0 \bar{t}_0 \quad (35)$$

Daraus ergibt sich für die Lage der Eintrittspupille:

$$\bar{t}_0 = \frac{\bar{y}_1}{\bar{u}_0} \quad (36)$$

Die Größe kann anhand des Abbildungsmaßstabs errechnet werden:

$$M_{entp} = \frac{n_0 u_0}{n_j u_j} = \frac{n_{as} \bar{u}_{as}}{n_0 \bar{u}_0} \quad (37)$$

$$r_{entp} = M_{entp} r_{as}$$

Analog wird auch die Position und Größe der Austrittspupille berechnet. Man projiziert die Aperturblende in die Objektebene:

$$\bar{y}_{j+1} = \bar{y}_j + \bar{u}_j \bar{t}_j = 0 \quad (38)$$

und somit

$$\bar{t}_j = -\frac{\bar{y}_j}{\bar{u}_j} \quad (39)$$

Die Größe kann auch anhand des Abbildungsmaßstabes errechnet werden:

$$M_{extp} = \frac{n_0 u_0}{n_j u_j} = \frac{n_{as} \bar{u}_{as}}{n_j \bar{u}_j} \quad (40)$$

$$r_{extp} = M_{extp} r_{as}$$

Nicht nur die Strahlungsleistung, sondern auch das Sichtfeld eines optischen Systems wird durch diverse Engstellen im optischen System begrenzt. Das Element, welches das Sichtfeld limitiert, wird **Feldblende** genannt. Die Lage der Feldblende kann durch die Skalierung des Hauptstrahls bestimmt werden. Das Element des optischen Systems mit dem kleinsten Verhältnis $|r/\bar{y}|$ ist die Feldblende (field stop).

$$fs = \min_i \left| \frac{r_i}{\bar{y}_i} \right| \quad (41)$$

Die Abbildungen der Feldblende in der Objektebene und in der Bildebene werden entsprechend **Eintrittsluke** und **Austrittsluke** genannt. Um die Lage und die Größe von beiden Luken bestimmen zu können, sollen zwei neue Strahlengänge, für einen Axial- und einen Hauptstrahl mit $y'_{fs} = 0$ bzw. $y'_0 = 0$, durch das optische System gerechnet werden. Nach [OSH85] kann jeder Paraxialstrahl durch eine gewichtete Summe von einem axialen und einem Hauptstrahl dargestellt werden.

$$y'_m = \alpha y_m + \beta \bar{y}_m \quad (42)$$

Da die Strahlen beliebig skaliert werden können, nehmen wir an, dass $\alpha = 1$ ist. Dann ist der zweite Koeffizient:

$$\beta = -\frac{y_{fs}}{\bar{y}_{fs}}. \quad (43)$$

Betrachten wird nun die Objektseite. Das heißt, dass der Axialstrahlgang rückwärts geroutet wird:

$$y'_0 = y'_1 - u'_0 t'_0 = 0. \quad (44)$$

mit

$$y'_1 = y_1 + \beta \bar{y}_1 \quad (45)$$

$$u'_0 = u_1 + \beta \bar{u}_0$$

Daraus ergibt sich die Lage der Eintrittsluke:

$$t'_0 = \frac{y'_1}{u'_1} = \frac{y_1 + \beta \bar{y}_1}{u_1 + \beta \bar{u}_0} \quad (46)$$

Analog dazu kann die Lage der Austrittsluke in der Objektebene bestimmt werden:

$$t'_j = -\frac{y'_j}{u'_j} = -\frac{y_j + \beta \bar{y}_j}{u_j + \beta \bar{u}_j} \quad (47)$$

Durch Berechnung des Abbildungsmaßstabes können die Größen der Luken bestimmt werden:

$$M_{entp} = \frac{n_{fs} \bar{u}_{fs}}{n_0 \bar{u}_0}, \quad M_{extp} = \frac{n_{fs} \bar{u}_{fs}}{n_j \bar{u}_j} \quad (48)$$

Einer der wichtigsten Parameter für einen Entwickler des optischen Systems ist das **Sichtfeld**. Dieses gibt das Winkelmaß des maximalen Ausschnittes an, den das optische System abbilden kann:

$$\theta = 2 \arctan(-\bar{u}_0 \cdot \frac{r_{fs}}{\bar{y}_{fs}}) \quad (49)$$

Die nächste wichtige Größe, die anhand eines parallelen Strahls der paraxialen Berechnung abgeschätzt werden kann, ist die effektive Brennweite eines optischen Systems. Die effektive Brennweite ist definiert als Abstand zwischen dem Brennpunkt und der dazugehörigen Hauptebene des Linsensystems. Andersgesagt ist die effektive Brennweite gleich der Brennweite einer dünnen Linse, die das Mehrlinsensystem physikalisch ersetzen kann:

$$f_{eff} = -\frac{y_0}{u_j} \cdot \frac{y_j}{|y_j|}. \quad (50)$$

Zusammenfassend kann man die Analyse des optischen Systems im paraxialen Gebiet in die folgenden Schritte unterteilen [OSH85]:

1. Definition und Parametrisierung aller optischen Komponenten im Gesamtsystem: Abstände, Radien, Materialien etc.;
2. Berechnung des Durchgangs eines axialen Strahls (y_k und u_k);
3. Bestimmung von Lage und Größe der Aperturblende;
4. Berechnung des Durchgangs eines Hauptstrahls (\bar{y}_k und \bar{u}_k) durch die Mitte der Aperturblende;
5. Bestimmung von Lage und Größe der Feldblende;
6. Berechnung von Lage und Größe der Pupillen und Luken anhand der Vorwärts- bzw. Rückwärtsrechnung der Strahlenverläufe.

Bei der Konzipierung eines optischen Systems hat der Entwickler die Möglichkeit, diverse Systemgrößen zu variieren, um die gewünschten Eigenschaften zu erzielen. Zu diesen Größen zählen: Anzahl der Ebenen und Flächen j , Abstand zwischen den Ebenen bzw. Flächen t_k , Biegeradien der Flächen R_k , Brechzahlen der Medien n_k , Größe bzw. Durchmesser der optischen Komponenten und Blenden r_k , sowie der Flächentyp (refraktiv oder reflektierend).

Am Ende der paraxialen Berechnung stehen die wichtigsten Größen, wie die Lage und die Größe der Feld- und Aperturblenden, Pupillen und Luken, effektive Brennweite, Blendenzahl und Öffnungswinkel für weitere Analysen des Sternsensorsichtfeldes fest. Die lineare Approximation der Strahlengänge liefert keine direkte Information über die Qualität der Abbildung. Alle Sterne erscheinen in der Bildebene als ideale Punkte. In der Wirklichkeit verursacht Nichtlinearität diverse optische Fehler. Diese sind der Hauptgegenstand des nächsten Kapitels.

5.5.4 Nichtparaxiale Optik und optische Aberrationen

Die zentrale Aufgabe der nichtparaxialen Berechnung der Optik ist die Abschätzung der Einflüsse der nichtlinearen Anteile der Strahlengänge auf die Qualität der Abbildung. Die Abweichungen vom idealen Bild nennt man optische Abbildungsfehler oder Aberrationen. Man unterscheidet zwischen Farbfehlern und monochromatischen Fehlern. Letztere werden weiterhin in Bildschärfefehler und Bildlagefehler unterteilt. Eine vollständige Eliminierung von Aberrationen ist nicht möglich. In der Regel führt die Verbesserung eines Fehlertyps zur Verschlechterung eines anderen. Die Kunst in der Optikentwicklung ist es, ein balanciertes System zu schaffen. Eine solche Optimierung ist sehr zeitintensiv und sehr kostspielig. Je nach Anwendung werden nur bestimmte Aberrationen minimiert.

5.5.4.1 Optische Aberrationen

Der wesentliche Unterschied zwischen der paraxialen und der nicht linearen Abbildung ist, dass Strahlen, die ihren Ursprung in einem Punkt haben, nicht mehr in einem idealen Punkt abgebildet werden. Die Abbildung wird gestreut und radial etwas verschoben. Die nachfolgende Abbildung veranschaulicht diese Effekte. Hier sind drei Spot-Bilder zu sehen. Diese

entstehen beim monochromen Licht entweder durch axiale Verschiebung der Bildebene oder durch Änderung der Objektposition. Wenn elektromagnetische Strahlen bestimmter Wellenlängen propagiert werden, entstehen Spotbilder für einzelne Farben, üblicherweise für Blau, Grün und Rot.

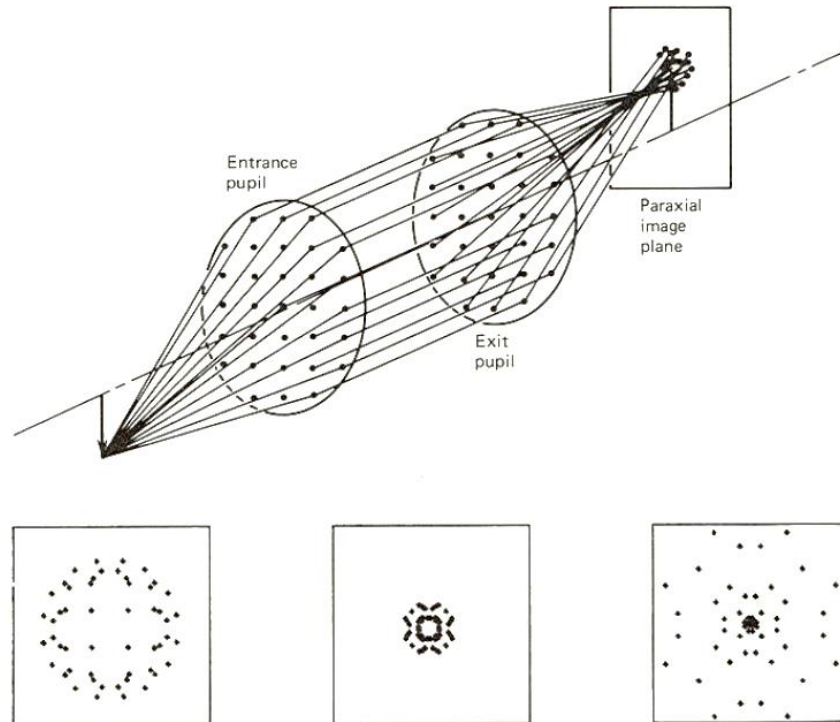


Abbildung 34. Entstehung der Spot-Bilder für drei verschiedenen Wellenlängen [OSH85]

Mit der Aberrationstheorie der optischen Abbildung haben sich berühmte Wissenschaftler wie F. Gauss, E. Abbe, L. Seidel, J. Petzval schon Mitte bis Ende des 19. Jahrhunderts beschäftigt. Sie erarbeiteten eine Darstellung der Strahlenabweichung in der Bildebene, mithilfe von Koeffizienten in Abhängigkeit von der idealen (anaxialen) Objekthöhe h und der Lage des Strahles in der Eintrittspupille, ausgedrückt in einem Polarkoordinatensystem mit Abstand s und Winkel θ . Die Seidelsche Fehlertheorie definiert fünf Typen der monochromatischen Aberrationen dritter Ordnung proportional zu s^3 , s^2h , sh^2 (zwei Typen) und h^3 : sphärische Aberrationen, Koma, Astigmatismus, Bildfeldwölbung sowie Verzeichnung [OSH85, PED08]. Die aufgelisteten Fehler werden als primäre Aberrationen bezeichnet. Die dazugehörigen Koeffizienten sind auch heutzutage in den Optical Design Tools für die Charakterisierung der Leistungsfähigkeit eines Systems weit verbreitet [LIT13]. Ein Nachteil einer solchen Beschreibung besteht darin, dass, sogar wenn die Aberrationen dritter Ordnung minimiert bzw. balanciert sind, die höheren Ordnungen zu dominieren beginnen und die optimale Lösung nicht erreicht wird.

Eine andere Charakterisierungsmethode basiert auf der direkten Analyse der Spotdiagramme [STA65, BRI78]. Das optische System wird anhand spezieller Kriterien, wie Radius des Spot-Kerns, Form des Spots und Schwerpunktposition relativ zur paraxialen Position, bewertet. Im

Fälle der Sternenabbildung sind diese Kriterien anschaulicher und praxisbezogener im Vergleich zu Aberrationskurven.

Diese Methode benötigt eine exakte bzw. nicht paraxiale Verfolgung von mehreren schrägen Lichtstrahlen durch das gesamte optische System. Dies ist mit Hilfe moderner Rechner relativ leicht realisierbar, wenn entsprechende theoretische Grundlagen vorliegen.

5.5.4.2 *Schrägstrahlverfolgung*

Die meisten Methoden für die Berechnung der schräg einfallenden Strahlen wurden bereits in den 60er und 70er Jahren entwickelt. Am weitesten verbreitet sind davon:

- Berechnung des Strahlendurchgangs mithilfe einer tangentialen Hilfsfläche nach Hopkins und Hanau [MIL62];
- Berechnung anhand einer Hilfstangente nach Feder [FED51, FED63, ZIM08];
- Analytische Methoden der sphärischen Geometrie nach Lebedev [SLJ69];
- Allgemeine Vektormethoden verfasst in [LIN17].

Die Berechnung der Strahlengänge nach Hopkins und Hanau ist allgemeingültig für sphärische und asphärische Flächen und ist gut dokumentiert. Weil diese Methode sehr anschaulich ist und leicht computerisiert werden kann, wird sie auch gerne in neueren Ausgaben der Lehrbücher zu optischem Design [MAL04, DER08] vorgestellt. Die Methode von Feder benutzt ähnliche und relativ einfache geometrische Auslegungen. Diese sind algebraisch und programmieretechnisch jedoch etwas schwieriger zu implementieren. Die Methode von Lebedev ist rein analytisch und hat nur wenig Bezug zu einem realen geometrischen Schema. Die Implementierung dieser Methode für Personen mit wenig Erfahrung ist sehr schwierig. Die von Psang Dain Lin verfasste Matrizen-Methode der Vektoralgebra ist sehr umfangreich und an den professionellen Anwender adressiert.

Im weiteren Verlauf dieses Kapitels wird der Schwerpunkt auf die Methode von Hopkins und Hanau gelegt. Es wird auf die Originalquelle [MIL62] zurückgegriffen. Um die Produktionskosten der zukünftigen Optik zu minimieren, werden nur sphärische Flächen in Betracht gezogen, denn für den Prototypenbau und kleine Serien ist die Produktion von asphärischen Linsen extrem kostspielig.

Der Durchgang eines Strahles wird in drei Schritten berechnet:

1. Zuerst wird der Übergang von der vorherigen optischen Fläche bis zur tangentialen Ebene der nächsten optischen Fläche berechnet.
2. Dann wird der Transfer zwischen der tangentialen Ebene und der dazugehörigen optischen Fläche bestimmt.
3. Zum Schluss wird der Durchgang durch diese sphärische Fläche kalkuliert.

Initiale Daten, die für die Berechnung des Strahlendurchgangs benötigt werden, sind:

- Koordinaten X_{-1}, Y_{-1}, Z_{-1} des Durchgangspunktes des Strahles durch die vorherige Fläche $j-1$.

- Die Richtung des Strahles, dargestellt als optischer Kosinus K_{-1}, L_{-1}, M_{-1} (siehe Abbildung 35):

$$K_{-1} = k_{-1} \cdot n_{-1}, L_{-1} = l_{-1} \cdot n_{-1}, M_{-1} = m_{-1} \cdot n_{-1} \quad (51)$$

- Es wird angenommen, dass die Z-Achse durch die Scheitelpunkte der Flächen $j-1$ und j verläuft. t_{-1} ist der Abstand zwischen den Scheitelpunkten der Flächen $j-1$ und j . n_{-1} und n sind die Brechzahlen der beiden Medien und c ist die Krümmung der sphärischen Fläche j .

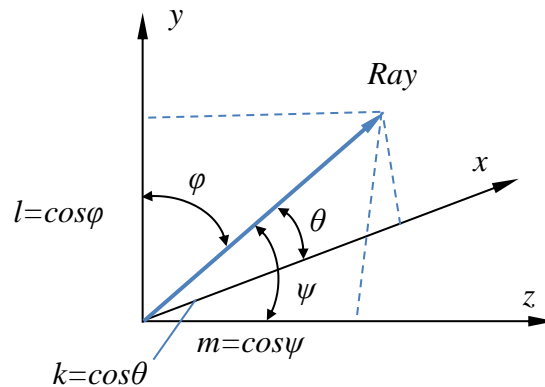


Abbildung 35. Darstellung der Strahlrichtung anhand der optischen Kosinus.

Abbildung 36 illustriert den Verlauf eines schrägen Strahls zwischen den Flächen $j-1$ und j . Da hier nur sphärische Flächen betrachtet werden, entfällt der Übergang zur Asphäre. Der Ursprung des Bezugskoordinatensystems befindet sich im Scheitelpunkt der zweiten optischen Fläche. Die Z-Achse stimmt mit der optischen Achse überein und ist in die Richtung der Bildebene ausgerichtet. Die Y-Achse verläuft orthogonal nach oben.

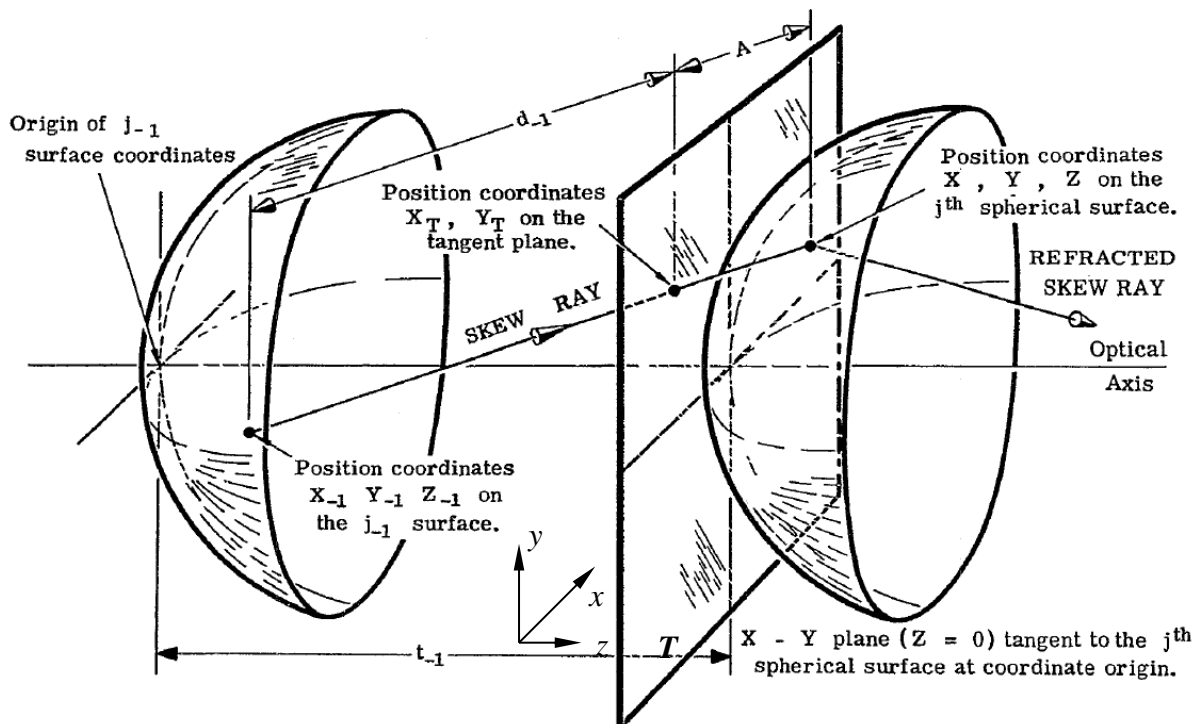


Abbildung 36. Strahlengang zwischen zwei optischen Flächen [MIL62].

- Berechnung des Schnittpunktes mit der tangentialen Fläche

Im ersten Schritt wird der Strahl vom Punkt (X_{-1}, Y_{-1}, Z_{-1}) bis zur tangentialen Ebene T verfolgt. Die Koordinaten vom Durchgangspunkt in dieser Ebene sind $(X_T, Y_T, 0)$. Diese Koordinaten werden relativ zu den vorherigen X_{-1}, Y_{-1} durch Addition der Änderungen ΔX und ΔY berechnet:

$$\begin{aligned} X_T &= X_{-1} + \Delta X = X_{-1} + d_{-1} \frac{K_{-1}}{n_{-1}}, \\ Y_T &= Y_{-1} + \Delta Y = Y_{-1} + d_{-1} \frac{L_{-1}}{n_{-1}} \end{aligned} \tag{52}$$

mit der Länge d_{-1} des Strahlenabschnittes von der Fläche $j-1$ bis zur tangentialen Fläche T .

Die Länge des Abschnittes d_{-1} ist nicht gegeben und muss bestimmt werden. Dafür wird die Projektion dieses Abschnittes auf die Z -Achse betrachtet. Der gleiche Abstand kann auch aus den initialen Daten berechnet werden. So werden die zwei folgenden Gleichungen erhalten:

$$\begin{aligned} \Delta Z &= d_{-1} \frac{M_{-1}}{n_{-1}}, \\ \Delta Z &= t_{-1} - Z_{-1}. \end{aligned} \tag{53}$$

Nun können die Gleichungen (53) nach $\frac{d_{-1}}{n_{-1}}$ aufgelöst werden und die Koordinaten des Schnittpunktes auf der tangentialen Ebene bestimmt werden:

$$\begin{aligned} \frac{d_{-1}}{n_{-1}} &= (t_{-1} - Z_{-1}) \frac{1}{M_{-1}}, \\ X_T &= X_{-1} + \Delta X = X_{-1} + d_{-1} \frac{K_{-1}}{n_{-1}} \\ Y_T &= Y_{-1} + \Delta Y = Y_{-1} + d_{-1} \frac{L_{-1}}{n_{-1}} \end{aligned} \tag{54}$$

- Berechnung des Schnittpunktes mit der sphärischen Fläche

Wenn der Schnittpunkt mit der tangentialen Ebene T berechnet wurde, kann man zum zweiten Schritt übergehen und den Strahl bis zur sphärischen Fläche propagieren. Der Strahlenabschnitt A ist kollinear zum Segment d_{-1} und hat somit den gleichen Richtungskosinus. Die Koordinaten (X, Y, Z) des Durchgangspunktes auf der sphärischen Oberfläche werden ähnlich wie zuvor bestimmt:

$$\begin{aligned} X &= X_T + \frac{A}{n_{-1}} K_{-1} \\ Y &= Y_T + \frac{A}{n_{-1}} L_{-1} \\ Z &= \frac{A}{n_{-1}} M_{-1} \end{aligned} \tag{55}$$

Jetzt muss nur noch die Länge des Abschnittes A ermittelt werden. Dieser ist abhängig von der Krümmung der j -ten Fläche, den Koordinaten des Durchgangspunktes sowie der Richtung des Strahls. Bei der Bestimmung des Abstandes A kann die Gleichung der Sphäre behilflich sein. Da der Mittelpunkt der sphärischen Fläche j der Punkt O mit den Koordinaten $(0,0,r)$ ist, nimmt die Gleichung der Fläche folgende Form an:

$$X^2 + Y^2 + (Z - r)^2 = r^2 \quad (56)$$

oder

$$X^2 + Y^2 + Z^2 - 2Zr = 0 \quad (57)$$

Wenn der Radius durch die Krümmung ($c = 1/r$) der Fläche ersetzt und die Gleichung modifiziert wird, ergibt sich:

$$c^2(X^2 + Y^2 + Z^2) - 2cZ = 0 \quad (58)$$

In die Gleichung können nun die Koordinaten des Durchgangspunktes der sphärischen Fläche eingesetzt werden:

$$c^2 \left(\left(X_T + \frac{A}{n_{-1}} K_{-1} \right)^2 + \left(Y_T + \frac{A}{n_{-1}} L_{-1} \right)^2 + \left(\frac{A}{n_{-1}} M_{-1} \right)^2 \right) - 2c \frac{A}{n_{-1}} M_{-1} = 0 \quad (59)$$

Durch Umformen der Gleichung folgt:

$$\left(\frac{A}{n_{-1}} \right)^2 c (K_{-1}^2 + L_{-1}^2 + M_{-1}^2) - 2 \left(\frac{A}{n_{-1}} \right) (M_{-1} - c(Y_T L_{-1} + X_T K_{-1})) + c(X_T^2 + Y_T^2) = 0 \quad (60)$$

Da die Summe des Richtungskosinus für einen Einheitsvektor gleich eins ist, vereinfacht sich der Koeffizient beim ersten Term zu cn_{-1}^2 . Die quadratische Gleichung insgesamt nimmt folgende Gestalt an:

$$cn_{-1}^2 \left(\frac{A}{n_{-1}} \right)^2 - 2B \left(\frac{A}{n_{-1}} \right) + H = 0, \quad (61)$$

mit

$$\boxed{\begin{aligned} H &= c(X_T^2 + Y_T^2), \\ B &= M_{-1} - c(Y_T L_{-1} + X_T K_{-1}) \end{aligned}} \quad (62)$$

Die Lösung der quadratischen Gleichung für $c \neq 0$ ist:

$$\frac{A}{n_{-1}} = \frac{B \pm n_{-1} \left(\left(\frac{B}{n_{-1}} \right)^2 - cH \right)^{\frac{1}{2}}}{cn_{-1}^2}. \quad (63)$$

Nur die negative Lösung ist interessant, da bei $c \rightarrow 0$ auch $A \rightarrow 0$ gehen soll. Außerdem sollen A und c gleiche Vorzeichen haben. Dies ist aus geometrischer Sicht sehr einfach darstellbar. Wenn die Krümmung c kleiner wird, nähert sich die sphärische Fläche der tangentialen

Ebene T an. Wenn $c > 0$ ist, handelt es sich um eine konvexe Fläche, A ist positiv und befindet sich vor der sphärischen Fläche. Wenn $c < 0$ ist, handelt es sich um eine konkave Fläche, A ist negativ und befindet sich hinter der sphärischen Fläche.

Abbildung 37 veranschaulicht das weitere Vorgehen bei der Bestimmung von A/n_{-1} . Um die Lösung weiter zu vereinfachen, wird die Einfallsebene des Strahles betrachtet. Die Einfallsebene verläuft durch den Strahlenvektor und den Mittelpunkt O der sphärischen Fläche.

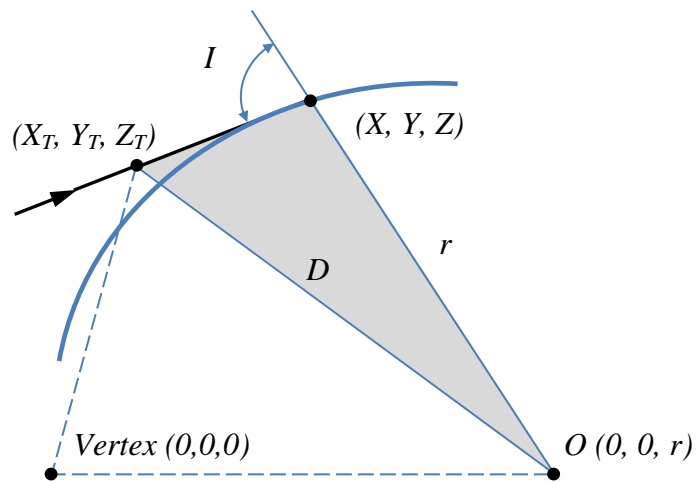


Abbildung 37. Durchgang eines Strahls durch sphärische Fläche [MIL62]

Nach dem Kosinus-Satz kann Abstand D aus Abbildung 37 folgendermaßen berechnet werden:

$$D^2 = A^2 + r^2 + 2Ar \cdot \cos(I). \quad (64)$$

Andererseits gilt:

$$D^2 = X_T^2 + Y_T^2 + Z_T^2 = X_T^2 + Y_T^2 + r^2. \quad (65)$$

Wenn man die beiden Gleichungen vereint und unter Berücksichtigung der Gleichung (62) für H nach $\cos(I)$ auflöst, ergibt sich:

$$n_{-1} \cos(I) = \frac{H - cn_{-1}^2 \left(\frac{A}{n_{-1}} \right)^2}{2 \frac{A}{n_{-1}}}. \quad (66)$$

Nun wird die Lösung (63) der quadratischen Gleichung (61) in die oben gelistete Gleichung (66) eingesetzt. Daraus ergibt sich folgender Zusammenhang:

$$n_{-1} \cos(I) = n_{-1} \left(\left(\frac{B}{n_{-1}} \right)^2 - cH \right)^{\frac{1}{2}}, \quad (67)$$

und umgestellt nach cn_{-1}^2 :

$$cn_{-1}^2 = \frac{B^2 - n_{-1}^2 \cos^2(I)}{H} = \frac{(B - n_{-1} \cos(I))(B + n_{-1} \cos(I))}{H}. \quad (68)$$

Wenn man erneut die Lösungsgleichung (63) für A/n_{-1} in Betracht zieht und die gerade abgeleiteten Ausdrücke für $n_{-1} \cos(I)$ und cn_{-1}^2 einsetzt, folgt die relativ übersichtliche Lösung:

$$\frac{A}{n_{-1}} = \frac{B - n_{-1} \cos(I)}{cn_{-1}^2}, \quad (69)$$

und letztendlich

$$\boxed{\frac{A}{n_{-1}} = \frac{H}{B + n_{-1} \cos(I)}}. \quad (70)$$

Die umrahmten Gleichungen sind die Kernformeln für die Berechnung der Transferstrecke zwischen zwei Flächen.

- Berechnung des Durchgangs durch die sphärische Fläche

Für die Berechnung eines Durchgangs durch die sphärische Fläche wird nun die Einfallsebene des schiefen Strahles in Abbildung 37 betrachtet. Die Aufgabe ist, den optischen Kosinus des gebrochenen Strahls zu finden. Dafür wird das Brechungsgesetz in Sinus- und Cosinus-Form verwendet:

$$n \cdot \sin(I') = n_{-1} \cdot \sin(I) \quad (71)$$

und

$$\boxed{n \cdot \cos(I') = n \left(\left(\frac{n_{-1}}{n} \cos(I) \right)^2 - \left(\frac{n_{-1}}{n} \right)^2 + 1 \right)^{\frac{1}{2}}}. \quad (72)$$

Der Durchgang des Strahls durch den Punkt (X, Y, Z) ist in Abbildung 38 illustriert.

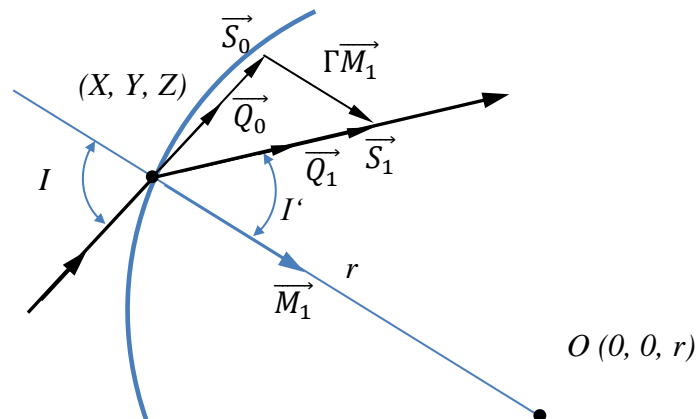


Abbildung 38. Brechung des Strahls an der sphärischen Fläche [MIL62]

Um den Richtungsvektor des gebrochenen Strahls bestimmen zu können, wird zuerst die Darstellung des Brechungsgesetzes in Vektorform benötigt. Seien \vec{Q}_0 und \vec{Q}_1 die Einheitsvektoren des Einfallsstrahls und des gebrochenen Strahls. Zusätzlich wird der Normaleinheitsvektor \vec{M}_1 in Richtung der Krümmung der Refraktionsfläche in Betracht gezogen. Berechnet wird jetzt das Kreuzprodukt der zwei Einheitsvektoren:

$$|\vec{Q}_0 \times \vec{M}_1| = |\vec{Q}_0| |\vec{M}_1| \sin(I) = \sin(I) \quad (73)$$

Gleiches gilt auch für das Kreuzprodukt $|\vec{Q}_1 \times \vec{M}_1|$. Dann sieht die Vektorform des Brechungsgesetzes wie folgt aus:

$$n|\vec{Q}_0 \times \vec{M}_1| = n_{-1}|\vec{Q}_1 \times \vec{M}_1| \quad (74)$$

Durch Ersetzen der Vektoren $n\vec{Q}_0$ und $n_{-1}\vec{Q}_1$ durch die Vektoren \vec{S}_0 und \vec{S}_1 ergeben sich die Gleichungen:

$$|\vec{S}_0 \times \vec{M}_1| = |\vec{S}_1 \times \vec{M}_1| \quad (75)$$

und

$$|(\vec{S}_1 - \vec{S}_0) \times \vec{M}_1| = 0 \quad (76)$$

Wenn das Kreuzprodukt zweier Vektoren gleich Null ist, sind diese Vektoren parallel. Das heißt:

$$\vec{S}_1 - \vec{S}_0 = \Gamma \vec{M}_1. \quad (77)$$

Durch die Projektion der Vektoren \vec{S}_0 und \vec{S}_1 auf den Vektor \vec{M}_1 folgt für Γ :

$$\Gamma = n \cdot \cos(I') - n_{-1} \cos(I). \quad (78)$$

Der Vektor \vec{M}_1 wird als Abstand zwischen dem Punkt O und dem Durchgangspunkt berechnet:

$$\vec{M}_1 = \frac{1}{r} \left((0 - X)\vec{i} + (0 - Y)\vec{j} + (r - Z)\vec{k} \right) = c(X\vec{i} + Y\vec{j} + (r - Z)\vec{k}). \quad (79)$$

Wenn die Länge des Vektors \vec{M}_1 bekannt ist, kann die Differenz der Vektoren \vec{S}_1 und \vec{S}_0 berechnet werden:

$$\vec{S}_1 - \vec{S}_0 = \Gamma \vec{M}_1 = -cX\Gamma\vec{i} - cY\Gamma\vec{j} + c(r - Z)\Gamma\vec{k}. \quad (80)$$

Andererseits werden die Vektoren \vec{S}_0 und \vec{S}_1 durch den optischen Kosinus repräsentiert:

$$\begin{aligned} \vec{S}_0 &= n_{-1}\vec{Q}_0 = K_{-1}\vec{i} + L_{-1}\vec{j} + M_{-1}\vec{k}, \\ \vec{S}_1 &= n\vec{Q}_1 = K\vec{i} + L\vec{j} + M\vec{k}. \end{aligned} \quad (81)$$

Die Differenz der beiden Vektoren beträgt:

$$\vec{S}_1 - \vec{S}_0 = (K - K_{-1})\vec{i} + (L - L_{-1})\vec{j} + (M - M_{-1})\vec{k}. \quad (82)$$

Durch Zusammenführen der beiden Gleichungen für $\vec{S}_1 - \vec{S}_0$ kann der neue optische Kosinus für den gebrochenen Strahl bestimmt werden:

$$\begin{array}{l} K = K_{-1} - cX\Gamma \\ L = L_{-1} - cY\Gamma \\ M = L_{-1} - (cZ - 1)\Gamma. \end{array} \quad (83)$$

Anhand der Gleichungen, die im Text mit Rahmen markiert sind, kann der exakte Strahlendurchgang bis zu einer sphärischen Fläche sowie durch diese Fläche berechnet werden.

5.5.4.3 Sonderfälle bei der Berechnung eines Strahlenverlaufs durch ein optisches System

Bei der Berechnung von Strahlengängen durch ein optisches System sollte man einige Situationen berücksichtigen, in denen ein Strahl nicht ordnungsgemäß propagiert werden kann und ggf. in den nachfolgenden Berechnungen nicht mehr berücksichtigt wird. Betrachtet wird ausschließlich die sequentielle Strahlenverfolgung. Dabei treffen die Strahlen Objekte oder Flächen in vordefinierter Reihenfolge. Das heißt, es werden keine Streulichteffekte erzeugt und berücksichtigt.

Abbildung 39 zeigt vier Sonderfälle, welche zur Terminierung der Berechnung eines Strahls führen. Solche Strahlen werden nicht mehr weiter propagiert.

Bei Totalreflektion ist der Brechungswinkel $I' > 90^\circ$, so kann der Strahl die Grenze zwischen zwei Medien nicht passieren und wird reflektiert. Dieser Effekt ist für eine Spiegeloptik charakteristisch, hat bei einer Linsenoptik jedoch negative Folgen.

Wenn die Krümmung der optischen Fläche relativ groß ist, kann es passieren, dass ein Strahl diese Fläche gar nicht treffen kann und daneben geht. Die weitere Berechnung ist in diesem Fall nicht sinnvoll.

Manchmal trifft ein Strahl ein physisches bzw. mechanisches Hindernis, sei es eine Blende, ein Linsenrand oder eine Fassung. Hier wird angenommen, dass ein Hindernis den Strahl vollständig absorbiert.

Linsen in einem optischen System haben einen bestimmten Außendurchmesser, der meistens durch den Herstellungsprozess bestimmt wird. Dieser Durchmesser begrenzt die maximale Durchgangshöhe der Strahlen. Die Strahlen können aus theoretischer Sicht weiter propagiert werden. Durch doppelte Totalreflektion an den „virtuellen“ Flächen wird die Ausbreitungsrichtung des Strahls nicht geändert und könnte unter Umständen nachfolgende Flächen treffen.

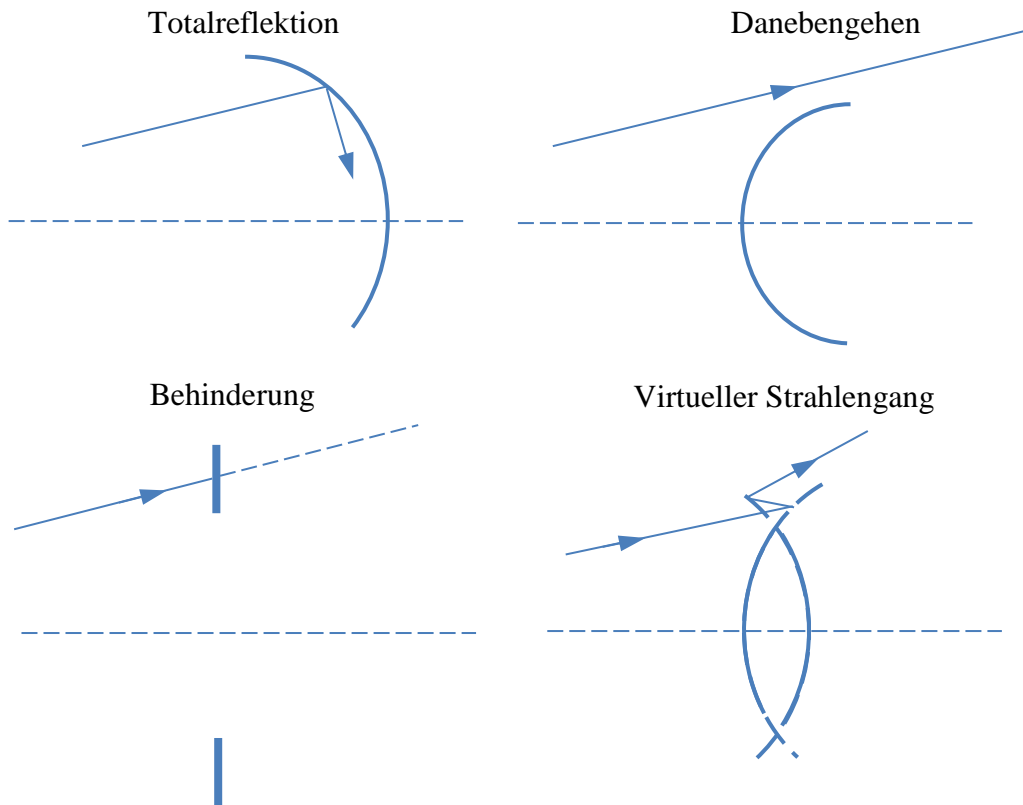


Abbildung 39. Sonderfälle der Strahlenverfolgung

5.5.4.4 Modell des nichtlinearen optischen Systems

Die Inputs und die Outputs, die für Berechnung der optischen Aberrationen notwendig sind, sind in Abbildung 40 dargestellt. Alle optischen Komponenten und Relationen zwischen diesen müssen vollständig spezifiziert werden. Die Anzahl der Linsen, deren Radien, Durchmesser, Dicken, Material und Abstände sind die essenziellen Inputs für die Berechnung der Strahlendurchläufe. Sie bestimmen die Größe und die Form des Spots. Die Reflektion und Absorptionseigenschaften der Linsen und der Linsenvergütung werden dabei nicht berücksichtigt.

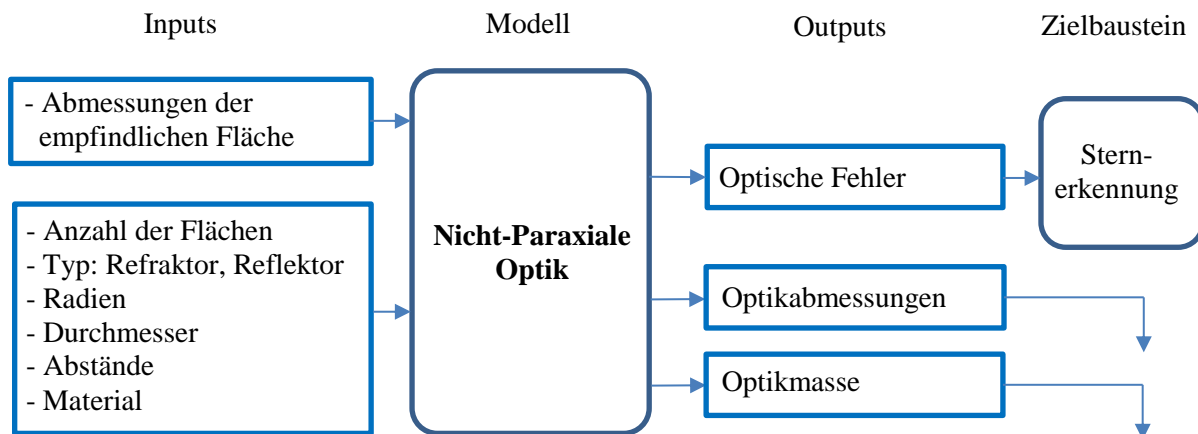


Abbildung 40. Nicht-Paraxiales Optikmodell

Parallel zur Evaluation des Optikdesigns anhand optischer Aberrationen können auch andere systemrelevante Eigenschaften des Gesamtsystems, wie z.B. die Masse und die Abmessungen, abgeschätzt werden.

5.5.5 Auflösung des gesamten opto-elektronischen Systems

Der ermittelte Wert für die Grenzmagnitude kann sich unter Einfluss von Umweltbedingungen verschlechtern. Dies passiert, wenn die Lichtenergie über eine größere Fläche als erwartet verteilt wird und das Sternbild sozusagen unscharf erscheint. Dafür kann es verschiedene Gründe geben. So kann z.B. eine Bewegung des Sternsensors während der Belichtung oder eine Änderung der geometrischen Größen von optischen Komponenten durch thermische Ausdehnung zur Vergrößerung des Spotbildes führen.

Wenn hingegen die Abbildung der Sterne sehr scharf ist und diese nur auf zwei bis drei Pixel abgebildet werden, sinkt jedoch die Genauigkeit von Positionsbestimmung im Subpixel-Bereich. In solchen Fällen wird die Optik absichtlich defokussiert.

Verschmierung und Defokussierung können begrenzt modelliert werden (siehe Abbildung 41). Die Schwierigkeit besteht darin, dass die Simulation von externen Faktoren wie der Satellitendrehrate abhängig ist.

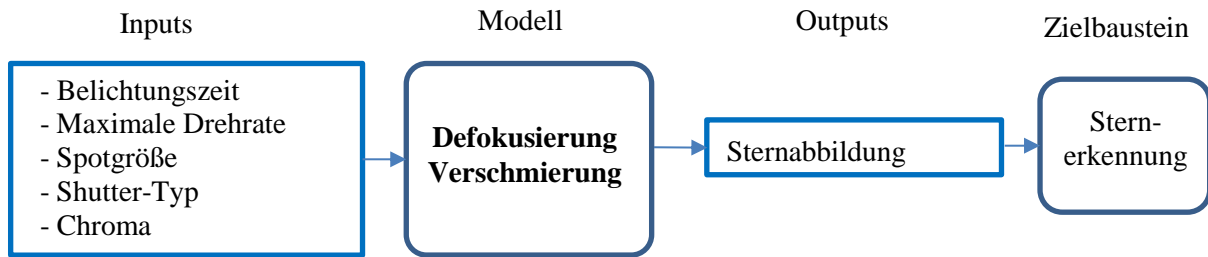


Abbildung 41. Modellierung der Sternform

5.5.5.1 Auflösungsvermögen eines optischen Systems

Die optische Auflösung eines Linsensystems bzw. eines Objektivs sagt aus, wie scharf feine Strukturen abgebildet werden können. Die theoretische Auflösung wird durch die Lichtbeugungsgrenze definiert. Bis jetzt wurde angenommen, dass sich das Licht nur geradlinig ausbreitet. In Wirklichkeit wird es an der Systemblende gebeugt. Anstatt einer Punktabbildung entsteht ein Beugungsscheibchen, auch als Airy-Scheibchen bekannt (siehe Abbildung 42). Es besteht aus einem hellen Zentrum und mehreren Interferenzringen mit abnehmender Helligkeit. Der Winkelabstand zwischen Abbildungen zweier Punktlichtquellen wird wie folgt berechnet:

$$\theta = 1,22 \cdot \frac{\lambda}{D} \quad (84)$$

mit λ – Wellenlänge des Lichts, D – Durchmesser der Aperturblende.

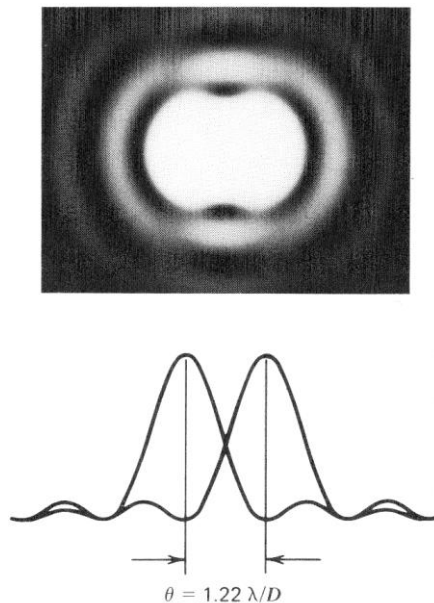


Abbildung 42. Abstand zwischen zwei Lichtpunkte [OSH85]

Aus Bequemlichkeit wird das Airy-Scheibchen oft durch eine Gauß-Verteilung approximiert. Die Interferenzringe werden hierbei vernachlässigt.

5.5.5.2 Approximation der Abbildung einer Punktlichtquelle

Die Helligkeitsverteilung einer Punktlichtquelle (Point Spread Function, PSF) kann als dreidimensionale Glockenkurve, auch Gaußkurve genannt, modelliert werden. Sie kann eine symmetrische oder asymmetrische Form haben und approximiert die Einflüsse der nicht linearen Optik auf die Abbildungsqualität. Die Form wird typischerweise experimentell für einen relativ hellen Stern (etwa ab 2^m) bestimmt. Abbildung 43 zeigt ein Beispiel mit der dreidi-

mensionalen Darstellung des Sterns Rigel (β Orionis 0,18^m). Die reale Aufnahme, links in der Abbildung, wurde vom Autor im Januar 2011 in Würzburg gemacht und anschließend mit einer Gaußglocke interpoliert. Danach kann das Modell auf Sterne anderen Klassen erweitert werden.

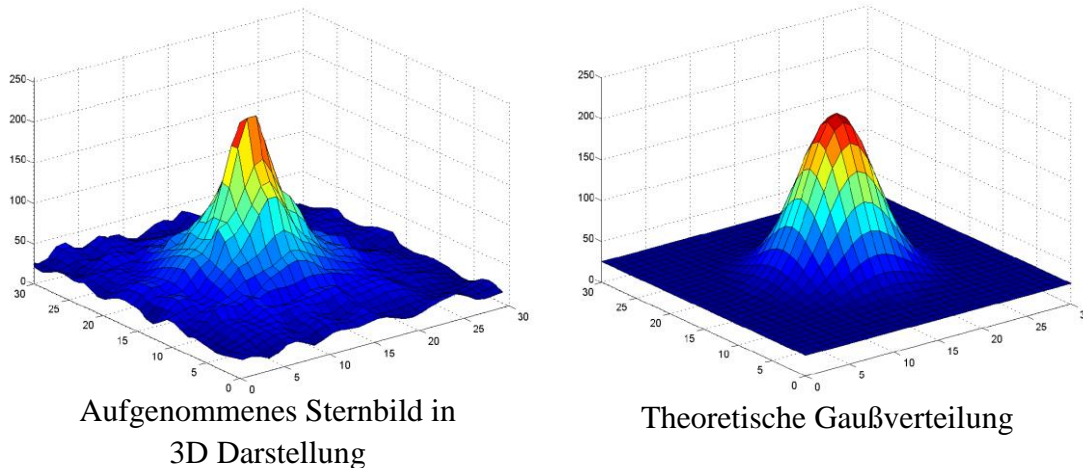


Abbildung 43. Aufgenommene und theoretisch nachgebildete Helligkeitsverteilung eines Sterns

Der Vorteil einer solchen Approximation besteht darin, dass Position und Magnitude relativ genau ermittelt werden können. Diese Darstellung hat aber auch einen gewissen Aufwand bei der Berechnung, da das mathematische Modell der Helligkeitsverteilung nicht linear ist:

$$f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\vartheta^2}} e^{\left(-\frac{1}{2(1-\vartheta^2)}\left(\left(\frac{x-\mu_1}{\sigma_1}\right)^2 - 2\vartheta\frac{x-\mu_1}{\sigma_1}\frac{y-\mu_2}{\sigma_2} + \left(\frac{y-\mu_2}{\sigma_2}\right)^2\right)\right)}, \quad (85)$$

mit

μ_i – Erwartungswerte,

σ_i – Standardabweichung,

ϑ – Korrelationskoeffizient.

Die approximierte Abbildung kann dann über das Pixelraster des Bildsensors gelegt werden, um die Grenzmagnitude ermitteln zu können [LIE02].

5.5.6 Verteilung der Sterne am Himmel

Um eine eindeutige Identifikation der Sterne im Bild durchführen zu können, muss sich immer, unabhängig von der Blickrichtung, eine ausreichende Zahl an Sternen im Sichtfeld des Sternsensors befinden. Je nach Identifikationsalgorithmus sind mindestens drei Sterne notwendig, um diese eindeutig identifizieren zu können. In der Praxis werden noch weitere Sterne benötigt, um die Algorithmen robuster gegen verschiedene Störungen zu machen. Für die Berechnung der Lage selbst sind bereits zwei korrekt ermittelte Sternvektoren ausreichend.

Sternsensoren mit einem größeren Sichtfeld (Field of View, FOV) können bei gleicher Öffnung und Grenzmagnitude mehr Sterne in einem Bild detektieren. Die Anzahl der sichtbaren Sterne beeinflusst direkt die Genauigkeit des Sensors. Je mehr Sterne in einer Berechnung verwendet werden, desto genauer ist die Messung [LIE02, MAR14].

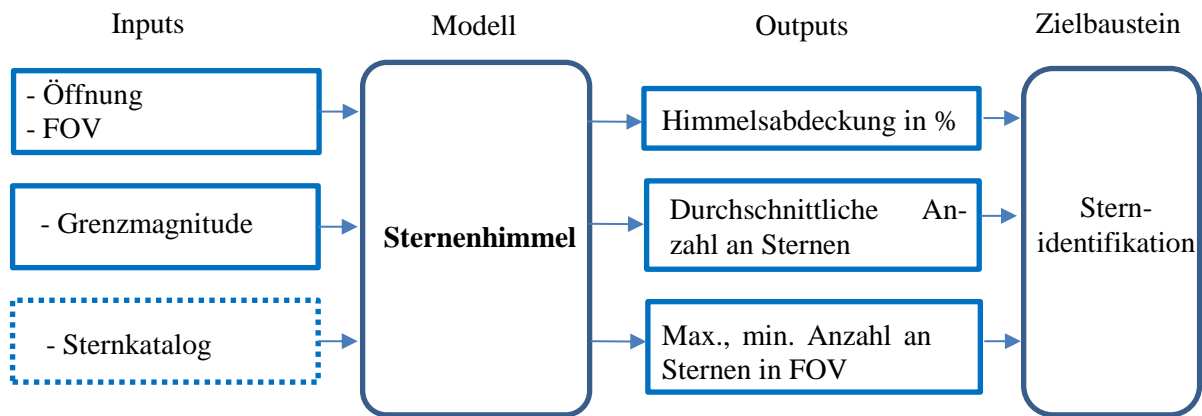


Abbildung 44. Modellierung des Sternenhimmels

Außerdem wird eine komplette Abdeckung des Sternenhimmels angestrebt, da die Identifikation in jeder Ausrichtung funktionieren soll. Auch die Verarbeitungsgeschwindigkeit ist von der Anzahl der Sterne abhängig. Eine größere Anzahl an zu verarbeitenden Sternen verlängert die Dauer der Auswertung. Daher wird der Sternenhimmel, wie in Abbildung 44 dargestellt, modelliert.

5.5.6.1 Vereinfachtes Modell des Sternenhimmels

Die durchschnittliche Anzahl an Sternen im FOV kann theoretisch sehr einfach berechnet werden. Für ein kreisförmiges FOV mit dem Öffnungswinkel α ist die Fläche des sichtbaren Sternhimmels:

$$S_{FOV} = 2\pi \left(1 - \cos \frac{\alpha}{2}\right). \quad (86)$$

Dabei wird vorausgesetzt, dass der Radius der Sternenhimmelssphäre $r = 1$ ist. Relativ zur gesamten Kugeloberfläche ($S = 4\pi$) ergibt sich dann ein Flächenanteil von:

$$\text{Anteil} = \frac{1}{2} \left(1 - \cos \frac{\alpha}{2}\right). \quad (87)$$

Die Gesamtzahl der Sterne in Abhängigkeit von der Grenzmagnitude m ist nachfolgend in Tabelle 6 zusammengefasst. Um die Berechnungen zu vereinfachen, wird die theoretische Annahme getroffen, dass die Sterne gleichmäßig über den gesamten Sternenhimmel verteilt sind.

Tabelle 6. Anzahl der sichtbaren Sterne in Abhängigkeit von der Grenzmagnitude

Grenzmag.	1	1,5	2	2,5	3	3,5	4	4,5	5	5,5	6	6,5	7	7,5	8
Anzahl der Sterne im Katalog	16	24	50	94	179	291	523	925	1637	2865	5068	8912	15588	25779	41475

Die Werte in Tabelle 6 steigen nahezu exponentiell. Durch die Interpolation mit einer e-Funktion gilt für den Zusammenhang zwischen Anzahl der Sterne im Katalog und Grenzmagnitude [LIE02]:

$$N = 6,75e^{1,08m}. \quad (88)$$

Daraus folgt für die durchschnittliche Anzahl an Sternen im FOV:

$$N_{FOV} = N \cdot \text{Anteil} = 6,75e^{1,08m} \cdot \frac{(1 - \cos \frac{\alpha}{2})}{2} = 3,375e^{1,08m} \left(1 - \cos \frac{\alpha}{2}\right). \quad (89)$$

Die Kurven in Abbildung 45 zeigen die durchschnittliche Anzahl an Sternen in einem kreisförmigen FOV für verschiedene Grenzmagnituden. So ist beispielweise ein Sichtfeld von min. 15° notwendig, um durchschnittlich mehr als 5 Sterne der 5. Magnitude zu sehen. Wenn man mit einem kleinen Sternkatalog von z.B. 300 Sternen arbeiten möchte, muss eine Optik mit dem Sichtfeld von etwa 25° eingesetzt werden, damit immer min. 3 Sterne im Sichtfeld sind.

Dieses vereinfachte Modell dient nur für eine erste Abschätzung der Systemparameter, weil die Geometrie des Detektors hier nicht berücksichtigt wird.

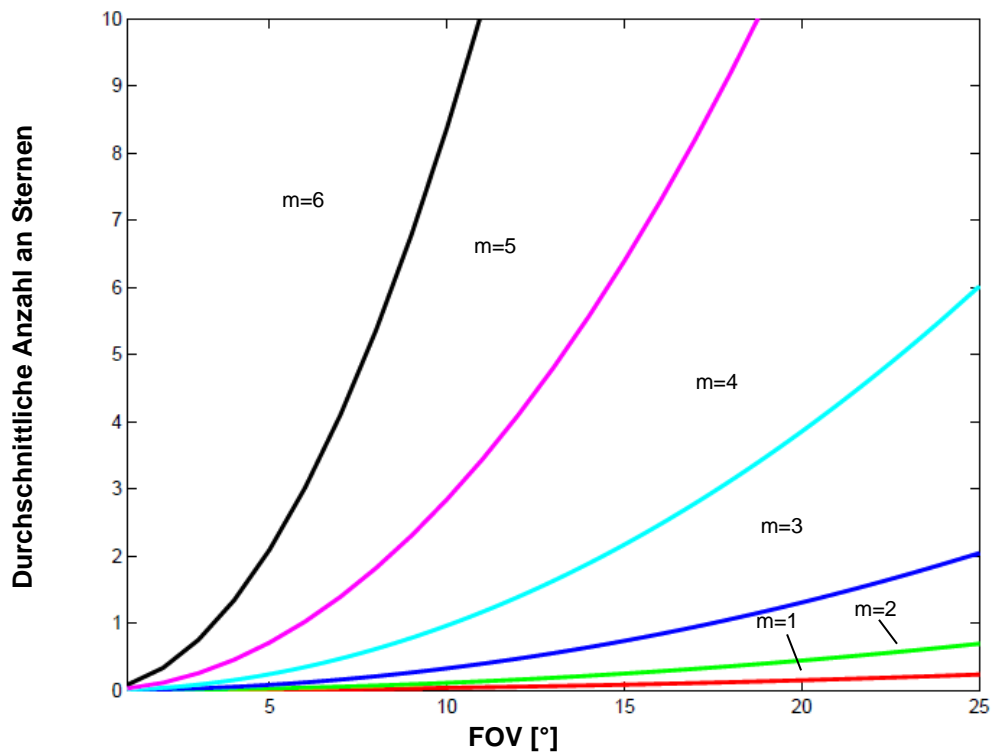


Abbildung 45. Durchschnittliche Anzahl an Sternen in einem kreisförmigen FOV

5.5.6.2 Exaktes Sternenhimmel-Modell

Da die Sterne nicht gleichmäßig über den realen Himmel verteilt sind und die meisten Detektoren bzw. Bildsensoren rechteckige Form haben, muss eine exakte Analyse der Sternenhimmelsphäre durchgeführt werden, um die Anforderungen an die minimale Sternenzahl im Bild bei der gewünschten Himmelsabdeckung (normalerweise 100%) zu erfüllen.

Um einen genauen Zusammenhang zwischen der Sternenzahl im Bild, Sichtfeld und Grenzmagnitude der Sterne finden zu können, muss eine Simulation durchgeführt werden, bei der der Sternhimmel mit einem Kamera-FOV mit der Dimension $H \times V$ abgescannt wird. Dies erfolgt mittels einer speziell dafür entwickelten Methodik [MOR04].

Um den Rechenaufwand zu minimieren, wird der vollständige Hipparcos-Sternkatalog [HIP20] zunächst auf die Sterne der vorgegebenen Grenzmagnitude reduziert.

Die Ausrichtung des Sternensensors wird durch den Vektor \vec{b} definiert, der durch die Mitte des FOVs verläuft. Um den gesamten Sternhimmel zu untersuchen, wird der Vektor \vec{b} um drei Achsen mit einer bestimmten Schrittweite gedreht. Das FOV selbst wird anhand von vier

Vektoren $\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4$ durch die Ecken des Bildsensors definiert (siehe Abbildung 46). Die Flächen zwischen den jeweiligen Nachbarvektoren begrenzen das FOV. Diese Flächen werden durch die vier Normalvektoren $\vec{n}_1, \vec{n}_2, \vec{n}_3, \vec{n}_4$ repräsentiert. Die Normalvektoren werden durch ein einfaches Kreuzprodukt von benachbarten \vec{b}_i -Vektoren kalkuliert und anschließend normiert.

Für jeden Stern aus dem reduzierten Katalog wird geprüft, ob er im FOV liegt. Dies geschieht durch skalare Multiplikation des normierten Vektors \vec{S}_i mit jeweils allen Vektoren \vec{n}_j . Dabei wird geprüft, ob alle Resultate größer oder kleiner als Null sind, d.h. ob der Vektor liegt außerhalb oder innerhalb des Sichtfeldes liegt.

$$\begin{aligned} \vec{S}_i \cdot \vec{n}_j > 0, \text{ Stern ist außerhalb des FOVs} \\ \vec{S}_i \cdot \vec{n}_j < 0, \text{ Stern ist innerhalb des FOVs} \end{aligned} \tag{90}$$

Als Ergebnis kann die Simulation des Sternenhimmels die minimale, maximale sowie durchschnittliche Anzahl an Sternen der Grenzmagnitude m im vorgegebenen Sichtfeld $H \times V$ eines Sternsensors liefern. Außerdem kann bei Bedarf eine komplette statistische Verteilung der Sternenanzahl erstellen werden. Ein Beispiel der Kalkulationsergebnisse für verschiedene Magnituden und verschiedene Sichtfelder ist in Tabelle 7 zusammengefasst. Ähnliche Ergebnisse wurden auch in den Artikeln [LIE93, KOL08] präsentiert.

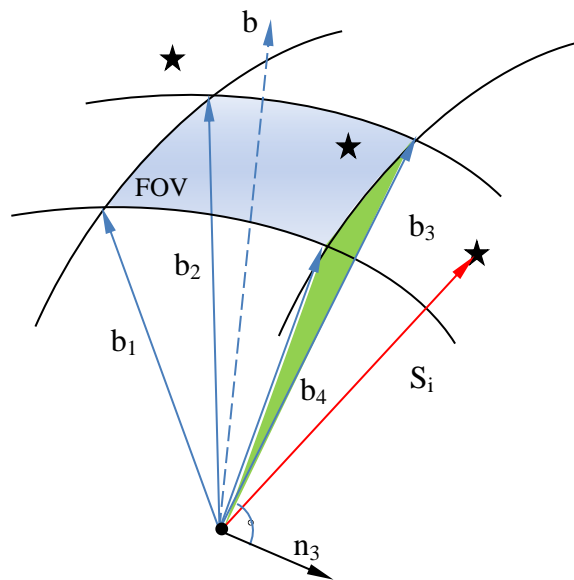


Abbildung 46. Bestimmung der exakten Anzahl an Sternen im FOV

Tabelle 7. Minimale und maximale Anzahl der Sterne in einem rechteckigen FOV

Sichtfeld H[°] x V[°]	Grenzmagnitude							
	4		5		5.5		6	
	min.	max.	min.	max.	min.	max.	min.	max.
5×5	0	4	0	8	0	11	0	19
10×10	0	9	0	19	0	32	1	46
15×20	0	18	1	43	5	70	14	107
20×20	0	22	2	50	9	83	21	135
20×25	0	26	2	63	13	100	27	162
30×30	0	37	9	90	28	148	56	246
45×45	7	60	33	150	73	272	136	443

Die existierenden Algorithmen zur Sternidentifikation nutzen für die eindeutige Bestimmung einer Sternkonstellation durchschnittlich 3 bis 15 Sterne. Um Überblendungseffekte des Sternsensors durch Sonnenstrahlung oder Reflexionsstrahlung der Erde zu minimieren, muss das FOV möglichst klein gehalten werden. Die Simulationsergebnisse zeigen, dass das optimale, theoretische Sichtfeld des Sternsensors mit einem quadratischen Bildsensor und einer Grenzmagnitude von etwa 5,5 etwa $17^\circ \times 17^\circ$ betragen könnte. Wenn der Bildsensor nicht quadratisch ist und das Seitenverhältnis z.B. 1,25 beträgt, sollte ein Sichtfeld mit $15^\circ \times 20^\circ$ ausreihend sein, um permanent mindestens 5 Sterne im Sichtfeld zu haben (siehe Abbildung 47).

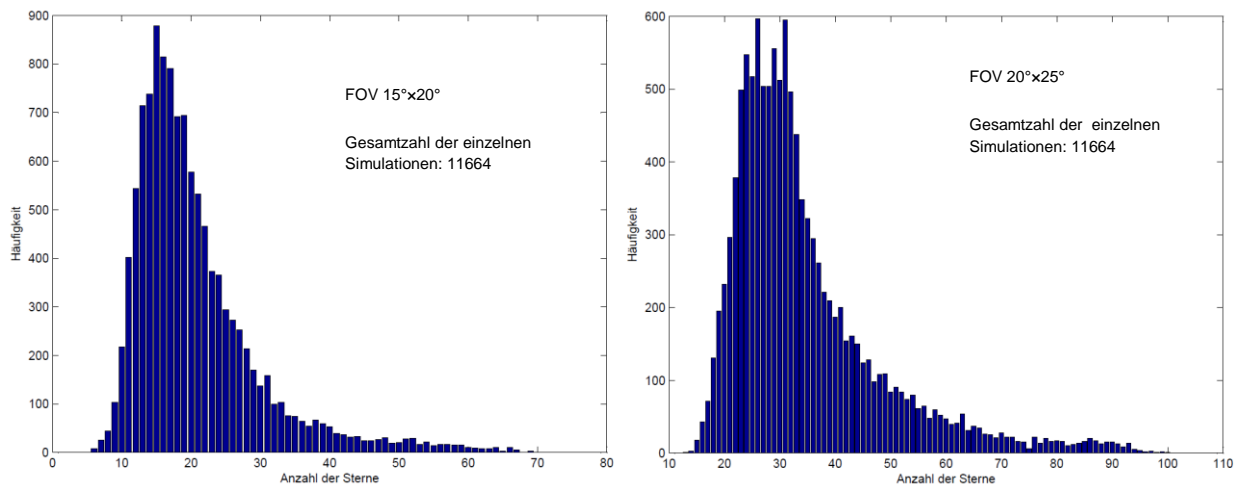


Abbildung 47. Häufigkeitsverteilung der Sterneanzahl im rechteckigen FOV

Bei dieser theoretischen Simulation anhand des Hipparcos-Starnkataloges [HIP20] wurde jedoch nicht berücksichtigt, dass ein reales Bild von verschiedenen Störfaktoren beeinflusst wird. Zu diesen Faktoren zählen Sensorrauschen, optische Fehler, Verdecken des Sichtfeldes durch die Erde und Überblendung durch die Sonne oder den Mond. All diese Faktoren verringern die Wahrscheinlichkeit, dass die Kamera eine ausreichende Anzahl an Sternen sieht. Aus diesem Grund wäre es sinnvoll, das Sichtfeld auf $20^\circ \times 25^\circ$ zu vergrößern. Abhängig von den Abmessungen der lichtempfindlichen Sensorfläche ergeben sich dadurch relativ kurze Brennweiten der Optik, sodass die Korrektur der optischen Aberrationen erschwert und die

Optik unnötig groß und teuer wird. Deshalb soll die FOV-Analyse nur unter realistischen Randbedingungen in jeder Entwicklungsiteration erneut durchgeführt werden.

5.5.7 Bild- und Datenverarbeitung

Dieses Kapitel fasst alle Softwarefunktionen bzw. -prozesse zusammen, die für eine einwandfreie Arbeit eines Sternsensors notwendig sind. Die Lagerfassung ist ein strukturierter, mehrstufiger Prozess und verläuft in mehreren nacheinander folgenden Schritten.

Es werden drei Gruppen von Prozessen unterschieden (siehe Abbildung 48). Die ersten zwei, primäre und sekundäre Prozesse, sind direkt als Module der On-Board-Sensorsoftware implementiert und laufen periodisch mit einer bestimmten Wiederholungsrate. Die primären Prozesse sind dominant und spielen die entscheidende Rolle bei der Erkennung und Identifikation der Sternmuster. Die sekundären Prozesse beeinflussen im Wesentlichen nur Performance-Parameter, wie Genauigkeit und Geschwindigkeit, und erhöhen die Zuverlässigkeit des gesamten Systems. Die dritte Gruppe der Prozesse wird normalerweise nur einmalig am Boden ausgeführt und die Ergebnisse der Berechnung werden im Sternsensor in Form von einer Datenbank nicht flüchtig abgespeichert.

Der erste Schritt im Arbeitsablauf eines Sternsensors ist die Aufnahme eines Bildes vom Sternenhimmel. Als nächstes werden, wenn Bedarf zur Verbesserung der Bildqualität besteht, Bildverzerrungsmaßnahmen durchgeführt. Danach erfolgt anhand spezieller Methoden die Erkennung von hellen und auffälligen Objekten im Bild. Aus den detektierten Objekten werden die Sterne extrahiert und zur weiteren Verarbeitung an den Lost in Space (LIS) Algorithmus übergeben. Der LIS-Algorithmus berechnet die charakteristischen Parameter der Sternkonstellation, die für das Durchsuchen der Datenbank notwendig sind. Die Datenbank wird in der Regel vorher am Boden vorbereitet, um den Rechenaufwand an Bord zu reduzieren. In der Datenbank werden alle Kombinationen der charakteristischen Parameter aller Sterne im sichtbaren Magnitudenbereich gespeichert. Die Einträge in der Datenbank werden anhand von einem der astronomischen Sternkataloge erstellt.

Die LIS-Algorithmen und Suchmethoden sind sehr rechenintensiv und benötigen für die komplette Berechnung der Lage auf einen eingebetteten Prozessor üblicherweise einige Sekunden oder sogar Minuten. Um die Ausgaberate der Sternsensoren zu erhöhen, werden oft rekursive Algorithmen verwendet [SAM05]. Diese Algorithmen nutzen zusätzlich verfügbare Informationen über die Satellitenlage, z.B. aus dem vorherigen Durchlauf des LIS-Algorithmus, oder zusätzliche Daten von anderen Sensoren (Gyroskope, Magnetometer), um die Position der bereits identifizierten Sterne weiter zu propagieren. Wenn die momentanen und vorhergesagten Positionen der Sterne übereinstimmen, entfällt der Identifikationsschritt. Die Lage kann ohne eine umfangreiche Suche in der Datenbank auf Basis der bereits identifizierten Sterne berechnet werden.

Solange genug Sterne im Bild erkannt und identifiziert wurden, kann die Lage des Satelliten anhand der Sternvektoren bestimmt werden. Hierfür werden die Sternvektoren aus einem aufgenommenen Bild mit den dazugehörigen Vektoren aus dem Sternkatalog zu einer Lageinformation fusioniert. Die berechnete Lage wird dann auf Plausibilität geprüft und in der gewünschten Darstellungsform ausgegeben.

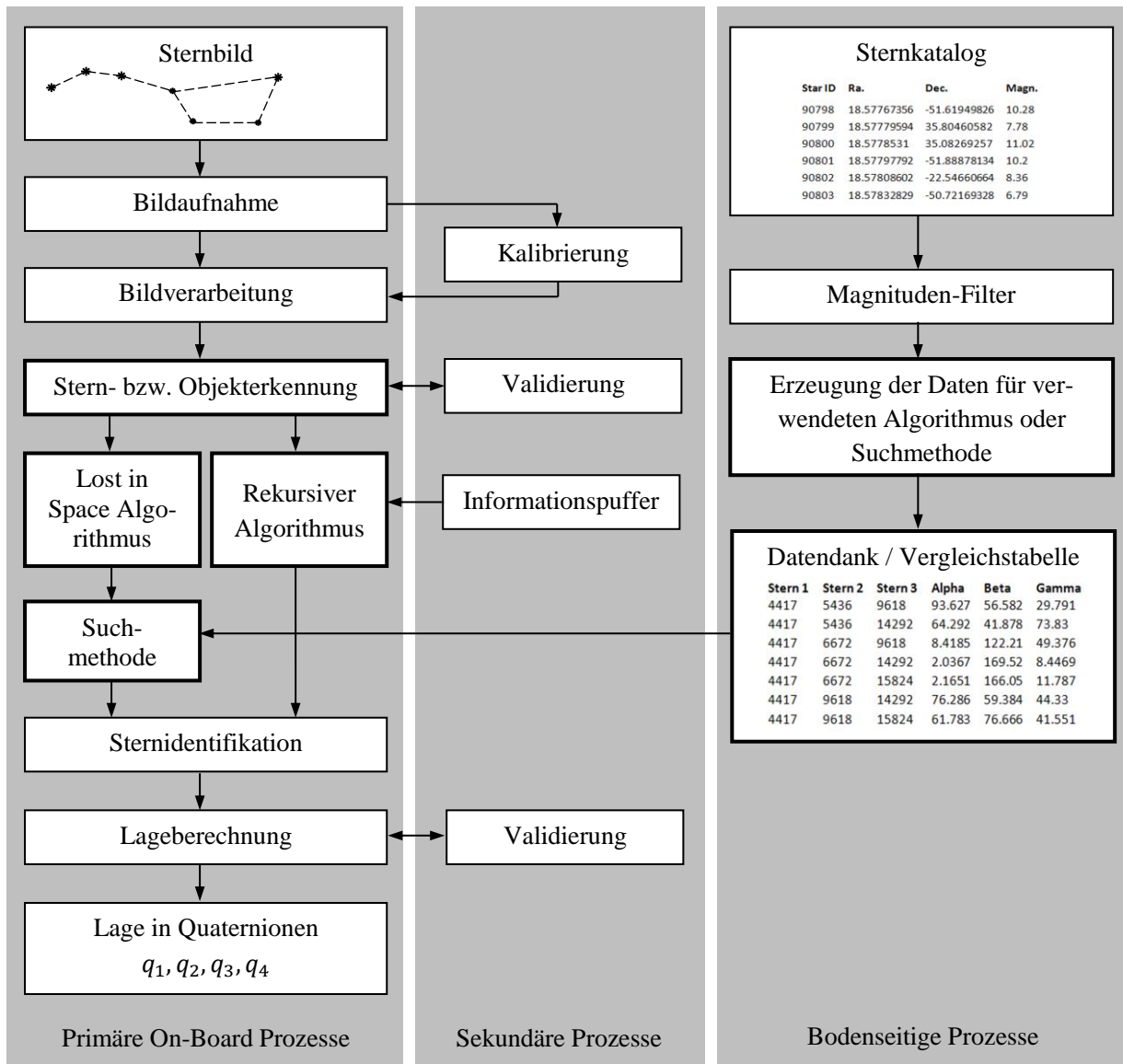


Abbildung 48. Struktur und Softwarekomponenten eines Sternsensors

Die beschriebenen Kernkomponenten der Softwarestruktur haben zwar streng definierte Aufgaben, können aber unterschiedlich realisiert werden. Um einen besseren Überblick über Vor- und Nachteile der existierenden Algorithmen und deren mögliche Implementierungen zu verschaffen, wird im weiteren Verlauf des Kapitels auf die wichtigsten Softwarekomponenten genauer eingegangen.

5.5.7.1 Objekt- und Sternerkennung

Die Objekterkennung hat die Aufgabe, alle leuchtenden Objekte im Bild zu erkennen. Unter leuchtenden Objekten versteht man nicht nur Sterne, sondern auch Streulichteffekte, Reflektionen, Planeten oder Satelliten, die nicht im Sternkatalog zu finden sind. Um darunter die Sterne bzw. Sternkonstellationen erkennen zu können, werden weitere vorhandene Bildinformationen, wie z.B. Größe und Anzahl der einzelnen Objekte, in Betracht gezogen. Abbildung 49 zeigt Objekt- und Sternerkennungsmodell mit entsprechenden Inputs und Outputs.

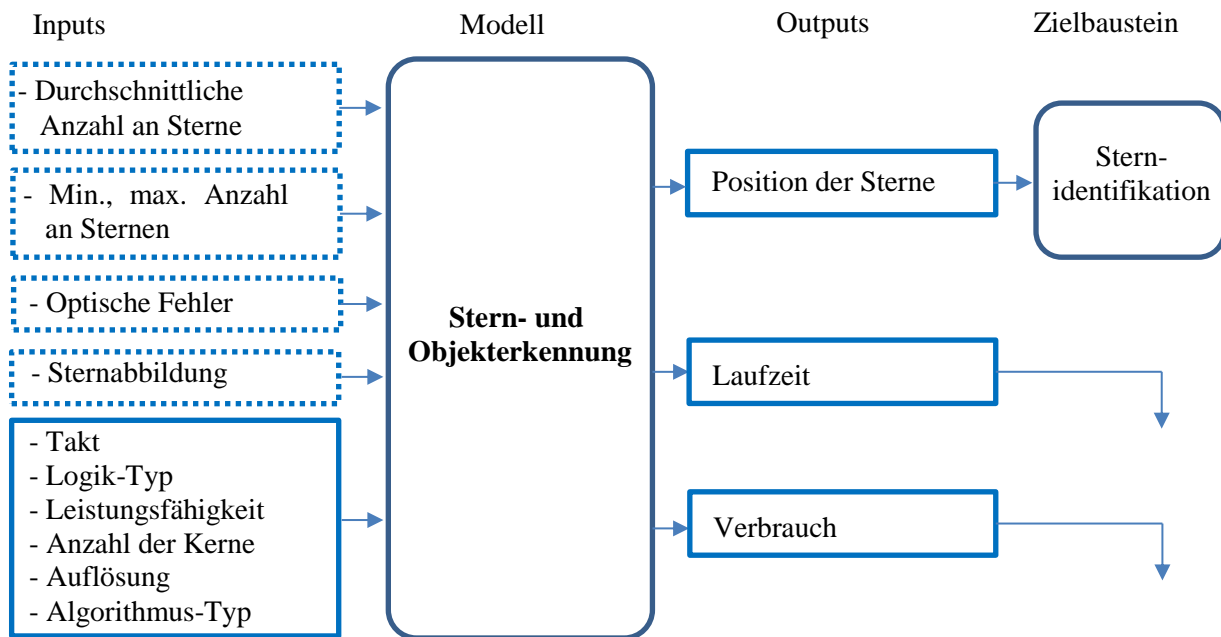


Abbildung 49. Modell der Sternerkennung

Um die Erkennung der Sterne zu erleichtern und die Messgenauigkeit zu erhöhen, wird die Optik einer Sternkamera in der Regel leicht defokussiert. In diesem Fall werden die Abbildungen der Sterne nicht in jeweils einem Pixel konzentriert, sondern auf mehrere Pixel verteilt und erhalten die typische Gaußverteilung der Helligkeit. Damit können die Sterne leichter von anderen Objekten unterschieden werden. Außerdem kann die Position der Sterne im Bild nun deutlich präzise im Subpixel-Bereich ermittelt werden. Die geometrische Position eines Sterns wird dann als Schwerpunkt (Centroid) der defokussierten Abbildung bestimmt.

Bei der Objekterkennung wird das digitale Bild Pixel für Pixel untersucht. Dabei wird ermittelt, welche Pixel mit ihrer Helligkeit über einem Schwellwert liegen. Wenn ein helles Pixel gefunden wurde, wird einer der folgenden Methoden verwendet, um die Position des Sterns zu bestimmen:

- Direkte Auswahl des hellsten Pixels

Jedes Pixel des aufgenommenen und gespeicherten Bildes wird mit einem vordefinierten Schwellenwert verglichen. Für ein Pixel, das heller als der Schwellenwert ist, wird geschaut, ob in der direkten Nachbarschaft hellere Pixel gibt. Es wird zum hellsten Pixel gewechselt und dann seine Nachbarn überprüft. Dieser Prozess wird solange wiederholt, bis ein lokales Maximum gefunden wurde. Als Sternposition wird der hellste Pixel in der Sternabbildung genommen. Die bereits untersuchten Bereiche des Bildes werden gelöscht und die Suche nach dem nächsten Stern fortgesetzt. Diese Methode ist nur mit Vergleichsoperationen und ohne weiteren rechnerischen Aufwand realisierbar. Sie ist zwar sehr einfach, dafür aber relativ ungenau. Die Genauigkeit ist durch die Pixelgröße begrenzt.

- Interessensbereich (Region of interest)

Bei dieser Methode wird zuerst ein Interessensbereich definiert, der untersucht werden soll. Der Bereich ist meistens quadratisch und kann eine statische oder variable Größe haben. Die

Größe ist so auszuwählen, dass äußere Flanken der Sterne gerade so erfasst werden und nicht zu viel Hintergrund mitberücksichtigt wird. Der Bereich wird entweder um das erste Pixel, das über dem Schwellenwert liegt, oder um das hellste Pixel in der Sternabbildung gelegt. In diesem Fall wird die Bereichsgröße oft proportional zur Intensität des hellsten Pixels ausgewählt.

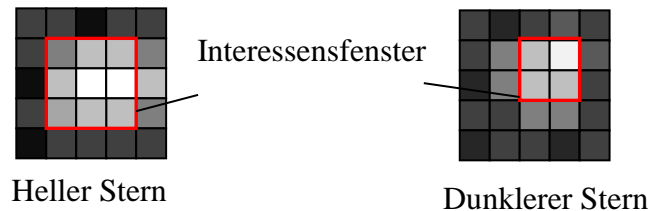


Abbildung 50. Ermittlung der Sternposition anhand eines Interessensfensters

Wenn ein Interessensbereich lokalisiert ist, werden die Koordinaten des Sterns als Schwerpunkt berechnet:

$$x_s = \frac{\sum_{x=1}^m \sum_{y=1}^n x \cdot h(x, y)}{\sum_{x=1}^m \sum_{y=1}^n h(x, y)}, \quad (91)$$

$$y_s = \frac{\sum_{x=1}^m \sum_{y=1}^n y \cdot h(x, y)}{\sum_{x=1}^m \sum_{y=1}^n h(x, y)}.$$

Hier ist $m \times n$ die Größe des Interessensbereichs und h ist die Helligkeit des jeweiligen Pixels in diesem Bereich. Der untersuchte Bereich wird gelöscht und die Suche nach einem nächsten Interessensbereich fortgeführt.

Wie zu erkennen ist, benötigt die Schwerpunktberechnung Fließkomma-Operationen, um Subpixelgenauigkeit zu erreichen. Durch die rechteckige Form des Interessensbereichs wird auch ein Teil des Hintergrunds in die Berechnung des Schwerpunkts miteinbezogen. Dies verschlechtert unter Umständen die Genauigkeit der Lageberechnung.

Die Methode der Interessensbereiche wurde speziell für die Sternerkennung entwickelt, kann aber in der Bildverarbeitung zu Bildsegmentierungsverfahren zugeordnet werden.

- Allgemeine Bildsegmentierung

Segmentierung ist ein Sammelbegriff für Verfahren zur digitalen Bilderverarbeitung. Dabei wird jedes einzelne Pixel des Bildes anhand verschiedener Kriterien einem bestimmten Bildsegment mit spezifischen Eigenschaften zugeordnet. Ziel ist es, das Bild in Bereiche mit bestimmten Eigenschaften zu unterteilen. Man unterscheidet zwischen Pixel-, Kanten-, Regionen- und Textur-basierte Verfahren der Bildsegmentierung. Der bekannteste Vertreter der Bildsegmentierungsalgorithmen auf Pixellevel ist der bereits mehrmals erwähnte Schwellenwert-Filter:

$$Pixel = \begin{cases} 1, & h \geq H \\ 0, & h < H \end{cases} \quad (92)$$

mit dem Helligkeitsschwellenwert H . Bei Sternsensoren liegt der Wert von H knapp über dem Bildrauschen.

Nach der Anwendung dieses Filters werden zwar zwei Bildsegmente, „Sterne“ und „Hintergrund“, eindeutig definiert. Die Pixel im Segment „Sterne“ hängen aber noch alle zusammen (Abbildung 51). Deshalb ist eine weitere Bearbeitung notwendig, z.B. mit Kanten- oder Regionenverfahren, um die Objekte voneinander zu separieren.

Die zusammenhängenden Pixel im Segment „Sterne“, die räumlich in der gleichen Region liegen, werden mit identischen Nummern markiert. Diese Markierung startet an einem beliebigen Pixel und propagiert weiter über die Nachbarn, bis alle einzelnen Objekte komplett erkannt sind. Geeignete Methoden dafür sind: Region-Growing, Split-and-Merge, Pyramid-Linking und Run-Length-Encoding sowie Konturen-Verfolgung [JÄH12, BUR15, NIS11].

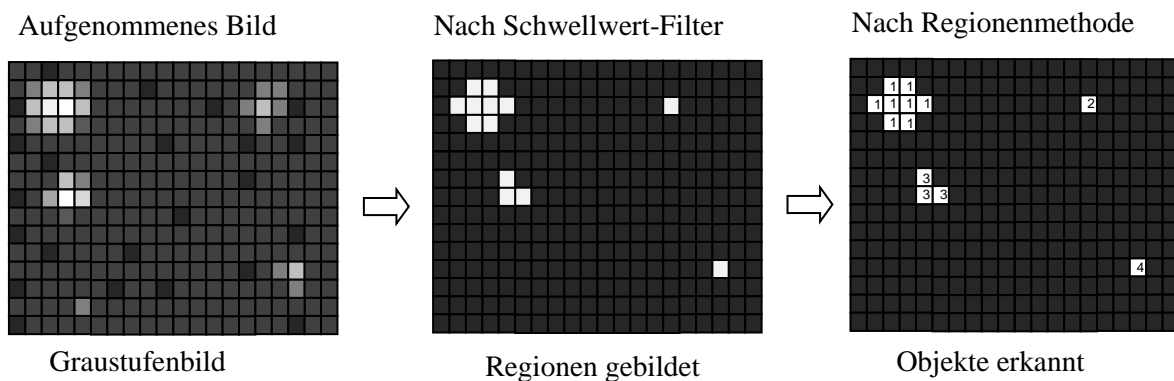


Abbildung 51. Regionenbasierte Bildsegmentierung

Bei der Bildsegmentierung werden sehr einfache logische Vergleiche durchgeführt, trotzdem wird mehr rechnerische Leistung benötigt, da diese Operationen mehrmals und für jeden Pixel des Sensors ausgeführt werden müssen. Außerdem muss bei manchen Methoden das vollständige Bild im Speicher abgelegt sein, um in der benötigten Reihenfolge auf die Pixel zugreifen zu können. Dies ist bei einfachen Hardwarearchitekturen nicht immer gegeben. Idealerweise soll das Segmentierungsverfahren das Bild direkt beim Auslesen aus dem Sensor verarbeiten. Deshalb wurde beim STELLA-Sternsensor eine Bildsegmentierungstechnik verwendet, welche die Speicherung von nur zwei Bildzeilen benötigt. Diese Methode wird in der Literatur als sequentielle Regionenmarkierung bezeichnet. Sie beginnt, wie in Abbildung 51 dargestellt ist, mit der Anwendung des Schwellwertfilters für jeden gelesenen Pixel. Wenn die Helligkeit eines Pixels e den Schwellenwert H überschreitet, wird eine Bildsegmentierungs-Operation ausgeführt, welche die Nachbarschaft nach bereits markierten Pixeln durchsucht (siehe Abbildung 52).

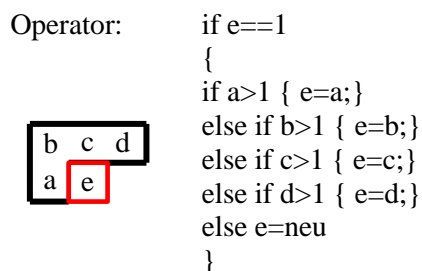


Abbildung 52. Bildsegmentierungsoperator der sequentiellen Regionenmarkierung.

Wenn keine von den Nachbarpixeln a, b, c, d bereits markiert ist, wird ein neues Objekt eingefügt. Wie man sieht, werden hier nur drei Pixel aus der vorherigen und ein aus der aktuellen Zeile untersucht. Diese Zeile kann beispielsweise im internen RAM-Speicher des Bildverarbeitungsprozessors abgelegt werden, sodass keine externen Speicherzugriffe notwendig sind.

Wenn alle Sternregionen markiert wurden, muss noch für jede Region dessen Schwerpunkt berechnet werden. Die einzelnen Schwerpunkte werden weiter als Sternkoordinaten in dem Identifikationsschritt verarbeitet.

Tests am realen Sternhimmel am Boden und im Orbit zeigen, dass die Methode der sequentiellen Regionenmarkierung gut für die Sternerkennung geeignet ist [BAL11-2, BAR13]. Lediglich die großen Sterne, deren Abbildung mehrere zehn Pixel groß ist, können unter Umständen in zwei bis drei Objekte zerteilt werden. Solche Sterne werden in der darauffolgenden Sternidentifikation nicht weiter berücksichtigt.

5.5.7.2 Berechnung der Magnitude

Zur Validierung der Sternerkennung werden manchmal Informationen über die Sternemagnitude benötigt. Die Magnitude der Sterne kann anhand folgender Formel berechnet werden:

$$M = \log_{2,512} \left(\frac{2,65 \cdot 10^{-6}}{E} \right), \quad (93)$$

$$E = k \cdot \sum_{i=1}^{Starpixel} h(i),$$

mit k – experimenteller Umrechnungsfaktor von Pixelhelligkeit in Leuchtstärke.

Der Faktor k wird experimentell bei der Kalibrierung des Sternsensors ermittelt. Dafür werden mehrere Sterne mit bekannter Magnitude aufgenommen. Die gesamte Pixelhelligkeit jedes Sterns und dessen Magnitude werden in die oben angegebenen Formeln eingesetzt und das Endergebnis für k gemittelt.

Informationen über die Sternemagnituden, die aus einem Bild errechnet wurden, können aber nur mit Vorsicht verwertet werden. Es gibt mehrere Faktoren, die das Ergebnis verfälschen können. Vor allem verursacht die Vignettierung des Lichtes durch das optische System einen Helligkeitsabfall am Rand des Bildes. Auch die Weltraumstrahlung und diverse Ausgangeffekte der verwendeten Materialien im Vakuum können die optischen Gläser vertrüben. Daher ist es empfehlenswert, die Magnituden für die Sternidentifikation nicht zu verwenden.

5.5.7.3 Korrektur der Abbildungsfehler (Kalibrierung)

Die Sternabbildung wird mittels eines optischen Linsensystems auf einem Bildsensor erzeugt. Beim Durchgang der Strahlen durch die Linsen treten verschiedene optische Fehler auf, die bereits in Kapitel 5.5.4 erläutert wurden. Die optischen Fehler verursachen eine Verzerrung des Gesamtbildes, bedingt durch die Ungleichmäßigkeit der Abbildungsmaßstäbe. Auch mechanischen Montagefehler des Sensors führen zu zusätzlichen Ungenauigkeiten bei der Lagebestimmung der Sterne. Unter Montagefehlern versteht man: Fokusfehler ΔF , Verschiebung des Mittelpunktes d , Verdrehung α und Neigung β . Diese sind in Abbildung 53 schematisch dargestellt.

Optische und mechanische Fehler können mithilfe polygonaler oder anderen nicht linearer Funktionen in Abhängigkeit von den Koordinaten der Bildpunkte korrigiert werden [RIC10]. So wird die Verzeichnung oft durch das Brown-Conrady-Modell beschrieben. Die Inversion dieser Beschreibung wird oft für die Korrektur der Abbildungsfehler, wie Verzeichnung, verwendet. Dieser Korrekturprozess wird auch Kalibrierung genannt.

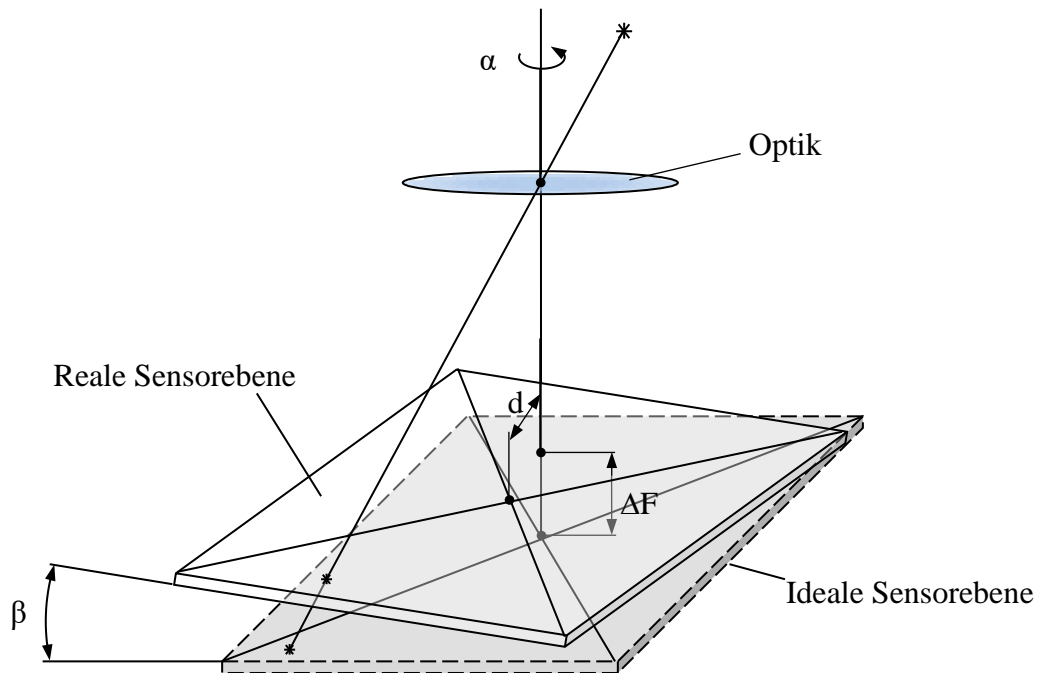


Abbildung 53. Montagefehler eines Bildsensors [LIE02]

Die Kalibrierung wird experimentell unter bestimmten Umweltbedingungen mittels eines Referenzbildes durchgeführt. Das Ergebnis wird entweder in tabellarische Form oder in Form von Koeffizienten einer mathematischen Funktion an Bord des Sternsensors gespeichert. Häufig werden die am Boden ermittelten Koeffizienten im Orbit etwas nachkorrigiert.

5.5.7.4 Validierung der Objekterkennung

Wenn alle hellen Objekte in einem aufgenommenen Bild erkannt sind, bedeuten es nicht automatisch, dass nur Sterne detektiert wurden. Auch Pixelfehler, vorbeifliegende Satelliten oder Weltraumschrottteile, Planeten und Lichtreflektionen können detektiert werden und die Lagebestimmung verfälschen. Es wird versucht, anhand der aus dem Bild extrahierten Hilfsinformationen, wie Objektgröße, Objekthelligkeit, Bewegungsprofil, solche Störungen zu entfernen. Diesen Schritt wird als Validierung der Objekterkennung bezeichnet und ist für jede Satellitenmission, sogar für jeden Sternsensor individuell zu implementieren.

5.5.7.5 Identifikation der Sterne

Wenn die Schwerpunkte aller Sterne im Bild berechnet wurden und die Korrektur der Abbildungsfehler durchgeführt wurde, kann mit der Identifikation der Sterne begonnen werden. An dieser Stelle kommen die Angaben zu den Sternpositionen am realen Himmel aus einem ast-

ronomischen Sternkatalog mit den gemessenen Sternvektoren zusammen und werden anhand spezieller Algorithmen verglichen.

Um die geforderte Ausgaberate gewährleisten zu können, muss die Sternidentifikation unter den vorhandenen energetischen Ressourcen extrem zeiteffizient arbeiten. Wie energie- und zeiteffizient ein Algorithmus arbeitet, ist von der Implementierung und selbstverständlich von der verwendeten Hardware abhängig.

Üblicherweise übernimmt ein Mikroprozessor oder Digital-Signal-Prozessor die Aufgabe der Sternidentifikation. Hier ist die Taktrate, Anzahl der parallel laufenden Cores bzw. Threads und die Leistungsfähigkeit der Floating-Point-Units essenziell.

Ein Sternkatalog bzw. eine Vergleichsdatenbank ist normalerweise in einem externen und nicht flüchtigen Speicher abgelegt. Abhängig von Speicher-Typ, -Architektur und -Schnittstelle kann der Sternkatalog unterschiedlich schnell gelesen werden. Auch die Größe der Datenbank, die direkt von der Anzahl der eingetragenen sichtbaren Sterne abhängig ist, spielt eine Rolle bei der Modellierung der Sternidentifikation (siehe Abbildung 54).

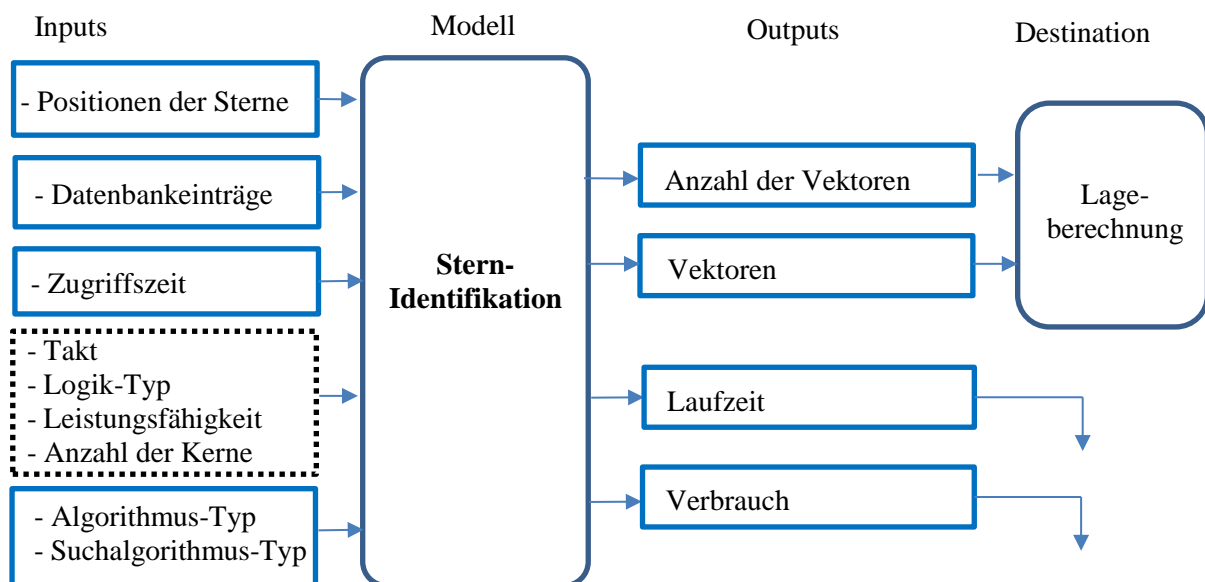


Abbildung 54. Model der Sternidentifikation.

- Lost-in-Space-Algorithmen

Nach erfolgreicher Durchführung der Sternerkennung und der nachfolgenden Korrektur der Montage- und Abbildungsfehler sowie Validierung stehen ausschließlich die Koordinaten der Schwerpunkte und eventuell die abgeschätzten Magnituden der Sterne aus dem aktuellen Bild zur weiteren Verarbeitung zur Verfügung. Es sind keine weiteren Informationen oder vorherige Messungen vorhanden. Eine solche Situation wird als Lost-in-Space bezeichnet. Der nächste Schritt ist die Identifikation des Sternmusters bzw. der Sternkonstellation im Bild. Dafür werden geometrische Merkmale der Konstellation betrachtet, die diese genau und eindeutig charakterisieren. Dies können verschiedene Abstände, Winkel oder sogar komplette geometrische Figuren sein. Diese einzigartige Kombination der Parameter wird mit Einträgen in der Datenbank verglichen und so nach einer passenden Sternkombination gesucht.

Um eine genaue Aussage über die Satellitenlage treffen zu können, müssen theoretisch nur zwei Sterne im Bild richtig identifiziert werden. In der Praxis werden jedoch viel mehr Sterne benötigt, um das Sternenmuster zu identifizieren. Die Anzahl der benötigten Sterne liegt im Durchschnitt zwischen 3 und 15 und ist vom verwendeten Algorithmus abhängig.

Generell unterscheidet man zwei Methoden der Sternidentifikation [HO12, NA06, SPR09]:

- **geometrische Methoden;**
- **Muster-Methoden.**

- Rekursive Algorithmen

Rekursive Algorithmen nutzen bereits verfügbarer Information, wie z.B. die Lage- bzw. Drehraten des Satelliten aus den vorherigen Erfassungen, um eine Vorhersage über die aktuelle Lage des Satelliten zu treffen. Die geschätzte Lage wird verwendet, um die Identifikation der Sterne im Bild zu beschleunigen. In diesem Fall ist das Durchsuchen der kompletten Datenbank nicht erforderlich. Es wird nur abgeglichen, ob die vorhergesagte Sternenkonstellation tatsächlich in FOV detektiert wurde.

Im weiteren Verlauf dieses Kapitels werden die bedeutendsten Methoden für LIS- Sternidentifikation ausführlich erläutert. Rekursive Verfahren werden im Rahmen dieser Arbeit nicht im Detail betrachtet.

5.5.7.6 Geometrische LIS-Methoden

Geometrischen Methoden bzw. Algorithmen nutzen direkt im Sternenbild messbare geometrische Parameter, wie die Abstände zwischen den Sternen (a, b, c), dreidimensionale Winkelabstände (φ_i), flache Winkel in der Bildebene (α, β, γ), Flächen (A), Flächenträgheitsmomente (I) oder eine Kombination zwischen diesen (siehe Abbildung 55). Die gleichen Parameter sind dementsprechend in der Referenzdatenbank zu finden.

Typische Vertreter der geometrischen Methoden sind:

- Dreiecksmethode,
- Pyramiden-Algorithmus,
- Polygon-Algorithmus,
- Gruppenzuordnungsmethode (Match Group Method) und
- Pol-Stern-Methode (Guide-Star).

Es existieren auch weitere Methoden, die im Grunde genommen nur eine Kombination oder Weiterentwicklung der oben genannten Methoden sind. Im weiteren Verlauf wird nur auf die vier wesentlichen und meistverbreiteten Methoden eingegangen.

- Dreiecksmethode

Die einfachste Methode zur Sternmustererkennung nutzt drei Sterne für Identifikation und wird deshalb Dreiecksmethode genannt. Um die Geometrie des Dreiecks beschreiben zu können wird eine Kombination aus drei Parameter verwendet.

Die grundsätzlichen geometrischen Merkmale eines Dreiecks, die aus einem aufgenommenen Bild berechnet werden können, sind:

a) Drei Winkelabstände $\varphi_1, \varphi_2, \varphi_3$ zwischen der Sternvektoren:

$$\begin{aligned}\varphi_1 &= \arccos \frac{\vec{b}_1 \cdot \vec{b}_2}{|\vec{b}_1| \cdot |\vec{b}_2|}, \\ \varphi_2 &= \arccos \frac{\vec{b}_2 \cdot \vec{b}_3}{|\vec{b}_2| \cdot |\vec{b}_3|}, \\ \varphi_3 &= \arccos \frac{\vec{b}_3 \cdot \vec{b}_1}{|\vec{b}_3| \cdot |\vec{b}_1|}.\end{aligned}\tag{94}$$

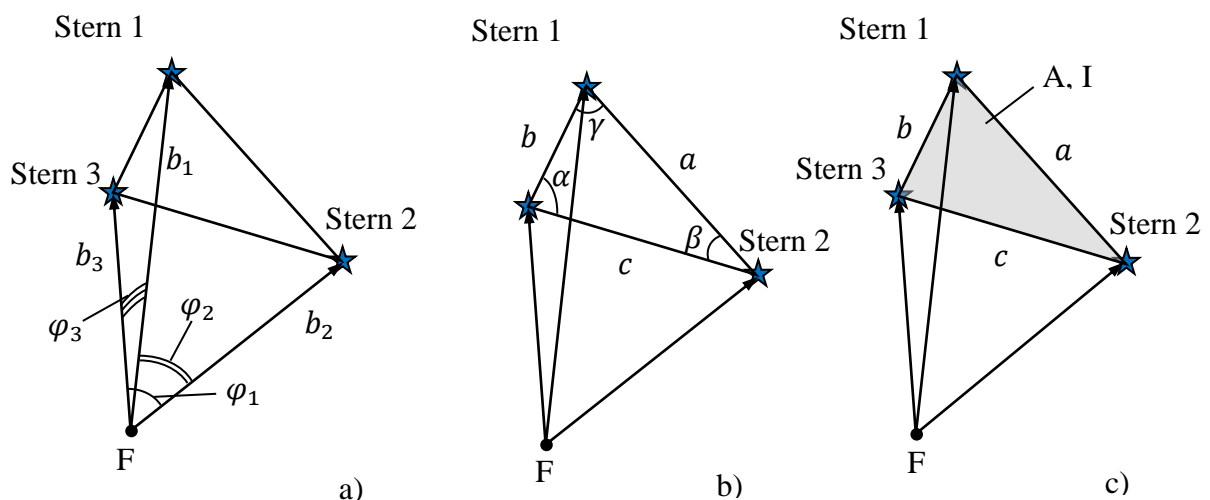


Abbildung 55. Geometrische Charakteristiken einer Kombination aus drei Sternen

Der Scheitelpunkt aller Winkelabstände ist der Brennpunkt F der Optik. Deshalb sind für die Berechnung genaue Informationen über die Brennweite der Optik und die Lage der Sensorfläche notwendig.

b) Drei Flächenwinkel α, β, γ zwischen den Sternen in der Dreiecksebene:

$$\begin{aligned}\alpha &= \arccos \frac{c^2 + b^2 - a^2}{2bc}, \\ \beta &= \arccos \frac{c^2 + a^2 - b^2}{2ac}, \\ \gamma &= \arccos \frac{a^2 + b^2 - c^2}{2ab}\end{aligned}\tag{95}$$

mit $a = |b_1 - b_2|$, $b = |b_1 - b_3|$, $c = |b_2 - b_3|$.

Die berechneten Flächenwinkel sind von der Brennweite unabhängig und reagieren weniger sensibel auf eine Temperatúrausdehnung der mechanischen und optischen Komponenten. Deswegen wird ein Verfahren auf Basis dieser Parameter auch als dimensionslos bezeichnet [SAM06]. Allerdings beschreiben die drei Flächenwinkel nur die Form des Dreiecks. Die Größe des Dreiecks wird dabei nicht berücksichtigt. Deshalb könnte es passieren, dass die eindeutige Identifikation nicht möglich ist, weil die Datenbank zwei gleiche Einträge hat, obwohl zwei unterschiedlich große Dreiecke beschrieben werden. Um dies zu vermeiden, kann eine weitere Größe, wie z.B. Fläche oder Flächenträgheitsmoment berücksichtigt werden.

c) Die Fläche und das polare Flächenträgheitsmoment des Dreiecks können wie folgt berechnet werden:

$$A = \sqrt{s(s-a)(s-b)(s-c)},$$

$$I = \frac{A(a^2 + b^2 + c^2)}{36}, \quad (96)$$

mit $s = (a + b + c)/2$.

Ein typischer Vertreter der geometrischen Dreiecksmethoden ist der Algorithmus von Liebe [LIE93, LIE95]. Hier werden als Kriterien zwei Winkelabstände φ_1, φ_2 zu den Nachbarsternen 1 und 2 sowie ein Flächenwinkel α verwendet.

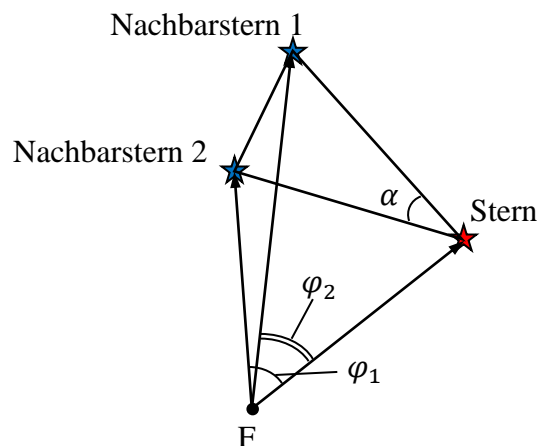


Abbildung 56. Geometrische Merkmale des Algorithmus nach Liebe [LIE93]

Nachdem die Winkel für drei aus dem Bild ausgewählte Sterne berechnet wurden, durchsucht man die Datenbank, um die am besten passende Kombination auszufiltern. In der Datenbank sind alle möglichen Dreieckskombinationen gespeichert, die Anhand eines Sternkatalogs unter Berücksichtigung des Kamera-FOVs gebildet wurden. Ein Beispiel einer solchen Datenbank für den Liebe-Algorithmus ist in Tabelle 8 dargestellt. Die Tabelleneinträge wurden mit den Faktoren 10, 10 und 0.2 multipliziert und als Ganzzahlen gespeichert. Die Anzahl der eingetragenen Dreiecke ist wesentlich größer als die Anzahl an sichtbaren Sternen. Für jeden Stern werden mehrere Dreiecke abgelegt. Diese sollen Unsicherheiten in der Winkelbestimmung und Variationen in der Magnitude abdecken.

Tabelle 8. Ausschnitt aus der Datenbank nach Liebe [LIE93]

Winkelabstand φ_1	Winkelabstand φ_2	Flächenwinkel α
...
54	80	32
54	80	33
54	81	32
54	81	33
...

Die Methode der „orientierten Dreiecke“ [LAM05] nutzt zur Identifikation unter anderem die Fläche des Dreiecks zwischen den drei nächsten Nachbarsternen. Für jeden sichtbaren Stern des Katalogs wird ein „orientiertes Dreieck“ gebildet. Die Orientierung wird anhand eines Pilotsterns ermittelt. Als Pilotstern wird ein Stern zwischen kürzester und längster Dreieckseite bezeichnet (siehe Abbildung 57). Der Pilotstern, die beiden Nachbarsterne, sowie ein vorzeichenbehafteter Flächenkoeffizient werden abgespeichert. Der Koeffizient wird als Betrag des Kreuzproduktes $|\vec{a} \times \vec{c}|$ ermittelt. Das Vorzeichen ist von der Lage des kurzen Vektors \vec{a} relativ zum Vektor \vec{c} abhängig. Das Vorzeichen ist positiv, wenn eine Drehung von \vec{a} zu \vec{c} im Uhrzeigersinn erfolgt.

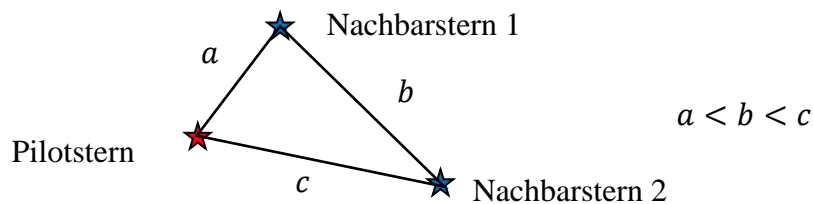


Abbildung 57. Orientiertes Dreieck nach Lamy au Rousseu [LAM05]

Nach der Bildaufnahme wird für jeden Stern ein orientiertes Dreieck berechnet. Unter Berücksichtigung der Messgenauigkeit werden passende Kandidaten mit identischen Flächenkoeffizienten aus der Datenbank extrahiert und in einer Liste abgelegt. Die Sterne, die am häufigsten in dieser Liste vorkommen, bilden die gesuchte Kombination.

Tabelle 9. Datenbank der Methode der orientierten Dreiecke [LAM05]

Stern	Pilotstern	Nachbarstern 1	Nachbarstern 2	Koeffizient
...
1258	13457	1258	23499	66,15
1315	4883	69948	1315	42,56
...

- Pyramiden-Algorithmus

Ist der Sensor empfindlich genug, um Sterne mit höherer Magnitude zu detektieren, wächst die Sternkataloggröße und somit die Anzahl der Einträge in der Datenbank nahezu exponentiell. Es kann vorkommen, dass einige Dreiecke in der Datenbank fast identische geometrische Charakteristiken haben. Diese können wegen der immer vorhandenen Ungenauigkeiten der Messung nicht mehr voneinander unterschieden werden. In diesem Fall ist keine eindeutige

Sternidentifikation mit den Dreiecksmethoden möglich. Um das richtige Dreieck von den anderen unterscheiden zu können, ist eine zusätzliche Information über die Sternkonstellation notwendig. Dies kann z.B. die Magnitude sein, wird aber selten verwendet, da die Messung sehr ungenau ist. Dieser zusätzliche Informationsbedarf wird im Pyramiden-Algorithmus durch die Einführung zusätzlicher Geometrien gedeckt [MOR04].

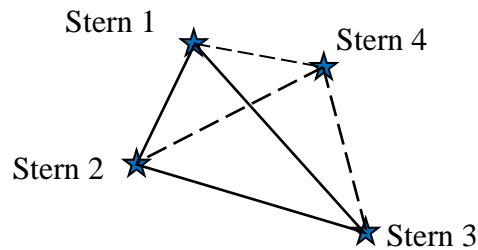


Abbildung 58. Sternkonstellation des Pyramiden-Algorithmus

Der Pyramiden-Algorithmus ist eine erweiterte Form des Dreieckalgorithmus und wird verwendet, wenn der Dreieckalgorithmus mehrere Dreiecke als richtig erkennt. Die eindeutige Lösung wird identifiziert, indem ein vierter Stern in die Berechnung einbezogen wird. Es entstehen drei zusätzliche Dreiecke, die weiter mit der Datenbank abgeglichen werden können (siehe Abbildung 58).

- Polygon-Algorithmen

Wenn der Pyramiden-Algorithmus weiterhin keine eindeutige Lösung liefert, werden noch weitere Sterne in Betracht gezogen, um das einzige korrekte Sternmuster zu finden [HER17].

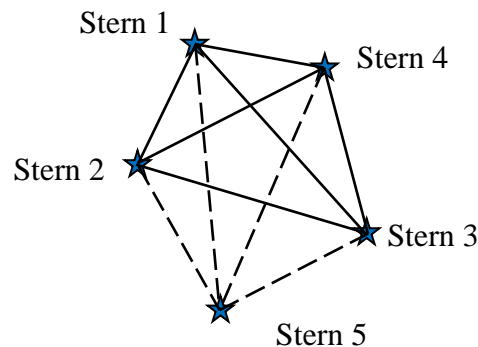


Abbildung 59. Sternkonstellation des Polygon-Algorithmus

In Abbildung 59 ist eine Sternkonstellation aus fünf Sternen dargestellt. Der fünfte Stern bringt weitere vier Winkelabstände und drei flachen Winkel für die Identifizierung der Muster mit. Allerdings müssen nicht alle geometrischen Parameter eines Musters bei der Identifikation verwendet werden. Dies ist bei der Pol-Stern-Methode der Fall.

- Pol-Stern-Methode (Guide-Star)

Eine weitere Methode für die Identifikation der Sterne ist die Pol-Stern-Methode [KOL08, KOS91, ROB05, LUO15]. Um diese Methode zu realisieren, wird zuerst ein Stern ausgewählt, der möglichst nah zur Bildmitte liegt. Dieser Stern ist der Pol des zukünftigen Musters. Nachdem der Pol ausgewählt ist, werden die Winkelabstände φ_i oder die flachen Winkel α_i

zur allen Nachbarsternen im Bild berechnet (siehe Abbildung 60) und mit entsprechender Datenbank verglichen (siehe Tabelle 10).

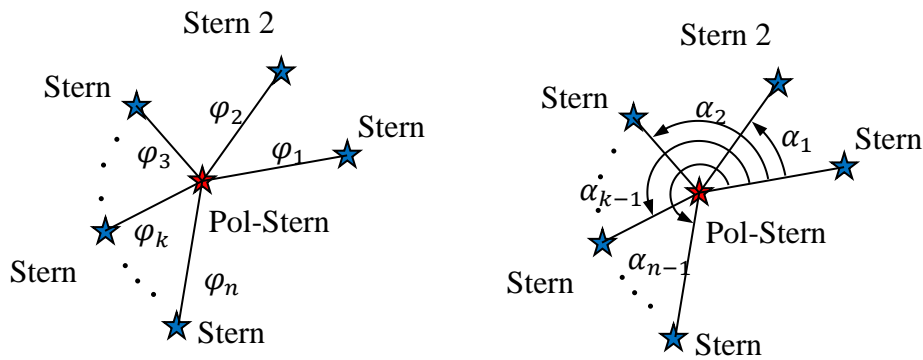


Abbildung 60. Sternmustercharakterisierung nach Pol-Stern-Methode

In diesem Fall enthält die Datenbank alle Winkelabstände von jedem Stern zu jedem Nachbarn, der theoretisch in ein Kamera-FOV hineinpassen könnte. Diese Abstände sind in der Regel nach Größe sortiert.

Tabelle 10. Datenbank nach Pol-Stern-Methode

Star-ID	φ_1	φ_2	φ_3	φ_4	...	φ_k	...	φ_n
676	7,41°	12,23°	13,56°	13,94°	...	17,35°	...	18,11°
744	4,27°	11,85°	17,24°	20,35°	...	-	...	-
1065	6,65°	12,45°	12,58°	19,12°	...	19,56°	...	19,89°
...

Wenn die Grenzmagnitude der sichtbaren Sterne bei einem Sternsensor mit einem diagonalen FOV von etwa 15° Magnitude 7 überschreitet, liegt die Größe einer solchen Datenbank im Megabyte-Bereich. Ein weiterer Nachteil dieser Methode ist, dass am Ende des Suchvorgangs nur ein Pol-Stern identifiziert ist. Um drei Sterne zu identifizieren, muss der Vorgang drei Mal wiederholt werden. Alternativ können die IDs von den Nachbarsternen zusätzlich gespeichert werden. Dies erlaubt es, die Nachbarn parallel zum Pol-Stern zu identifizieren. Dadurch verdoppelt sich aber unter Umständen die Datenbankgröße.

- Gruppenzuordnungsmethode (Match Group Method)

Diese Methode funktioniert ähnlich, wie ein Mensch die visuelle Erkennung eines Sternbildes durchführen würde [PAD97-2, ROE89, ROE92, ROE94, NA06]. Dazu wird die Magnitude der aufgenommenen Sterne, die bei der Sternerkennung kalkuliert werden muss, in die Identifikation miteinbezogen. Zuerst wird ein Set aus den N hellsten Sternen untersucht und die Abstände aller möglichen Paare berechnet. Die berechneten Abstände werden in der Datenbank gesucht und IDs der passenden Paare in einer tabellarischen Form zusammengefasst. Dann werden Gruppen gebildet, die einen gleichen Kernstern haben. Als Kernstern wird ein Stern bezeichnet, dessen ID am häufigsten in der Tabelle vorkommt, d.h. die meisten Übereinstimmungen mit der Datenbank hat. Die K Gruppen mit maximalen Übereinstimmungen werden beibehalten und alle anderen Gruppen werden eliminiert. Im nächsten Schritt werden die verbliebenen Gruppen validiert, indem man weitere Abstände zwischen den Gruppenmitgliedern betrachtet und die Genauigkeit im Vergleich zum Datenbankeintrag validiert (siehe Ab-

bildung 61). Die Gruppe mit maximaler Anzahl an Übereinstimmungen und präzisesten Abständen gewinnt und die restlichen Gruppen werden wieder verworfen.

Um die Identifikation erfolgreich durchführen zu können, benötigt dieser Algorithmus 8 bis 12 Sterne und ist somit nur für sehr empfindliche Sternensensoren oder für Sternensensoren mit einem größeren FOV geeignet. Der Datenbank ist im Vergleich zum Dreieckalgorithmus aber deutlich kleiner.

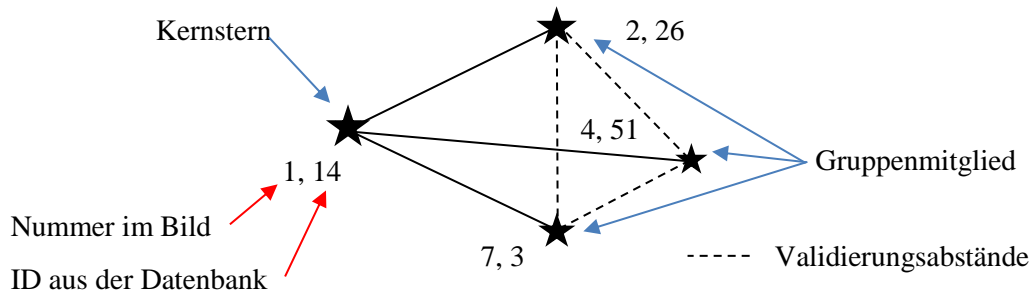


Abbildung 61. Match Group Method [ROE94]

5.5.7.7 Mustermethoden

Der Hauptunterschied der Mustermethoden im Vergleich zu den bisher beschriebenen geometrischen Methoden besteht darin, dass in einer Datenbank nicht die einzelnen geometrischen Parameter, sondern komplette Muster aller theoretisch im FOV sichtbaren Sterne abgelegt werden. Ähnlich wie Produkte in einem Supermarkt bekommt jeder Stern einen eigenen „Barcode“, der durch die Anordnung der Nachbarsterne bestimmt wird.

Als Beispiele für diese Technik werden folgende Methoden in Betracht gezogen:

- Gitter-Methode (Grid Method)
- Kreismuster-Methode
- Identifikation durch Neuronales Netz.

- Gitter-Methode

Im ersten Schritt dieser Methode [PAD97-1, PAD97-2] wird ein Referenzstern S_n im Zentralbereich des Bildes ausgewählt (siehe Abbildung 62 a)). Dann werden alle Nachbarsterne im Umgebungsbereich des Referenzsterns mit einem Radius r vorgemerkt. Die resultierende Sternkonstellation wird translatorisch so verschoben, dass der Referenzstern in der Mitte des Bildes liegt (siehe Abbildung 62 b)). Nun rotiert man alle Sterne solange um den Referenzstern, bis der nächste Nachbarstern waagrecht steht. Dann wird ein Gitter mit einem vordefinierten Raster mit $n \times n$ Zeilen über das Bild gelegt. Die Zellen werden von 0 bis $n \times n - 1$ nummeriert. Wenn der Schwerpunkt eines Sterns in einer Gitterzelle liegt, wird sie mit 1 (schwarz) markiert. Leeren Zellen bekommen dem Wert 0. Das so entstandene binäre Bitmuster wird entsprechend zu der Zellennummer codiert, z.B. ergibt sich aus Abbildung 62 d) für Stern S_n der Mustervektor (65, 93, 101, 158, 176). Genau dieser Vektor wird dann in der vorher angelegten Datenbank gesucht.

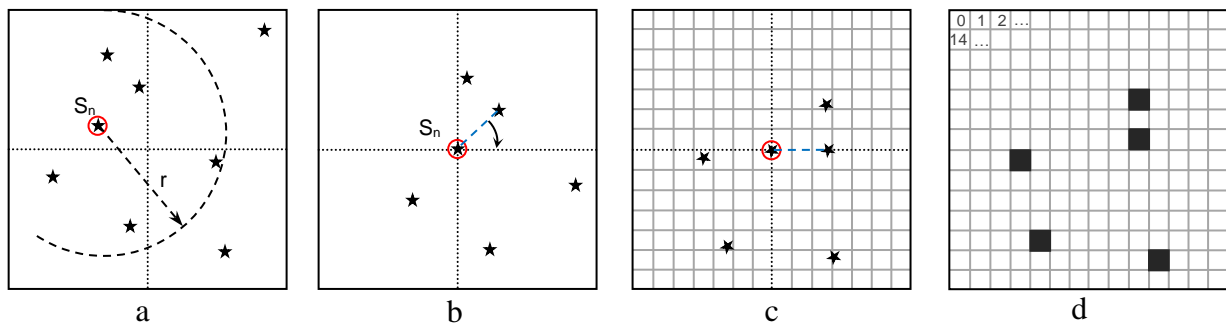


Abbildung 62. Gitter-Algorithmus [PAD97-1]

Beim Generieren der Datenbank aus einem Sternkatalog geht man genau die gleichen Schritte durch und berechnet für alle Sterne einen Mustervektor. Besonders interessant ist der Aufbau der Datenbank (Abbildung 63). Die Datenbank enthält $n \times n$ Zeilen. Im Beispiel in der Abbildung 62 besteht die Datenbank aus $14 \times 14 = 196$ Zeilen. Jede Zeile der Datenbank entspricht einer Gitterzelle und somit einem Bit im Muster. In jede Zeile werden die ID-Nummern der Referenzsterne eingetragen, bei denen das entsprechende Bit im Mustervektor als 1 markiert ist. Solche Datenbankorganisation bringt enorme Vorteile bei der Suche nach dem passenden Muster.

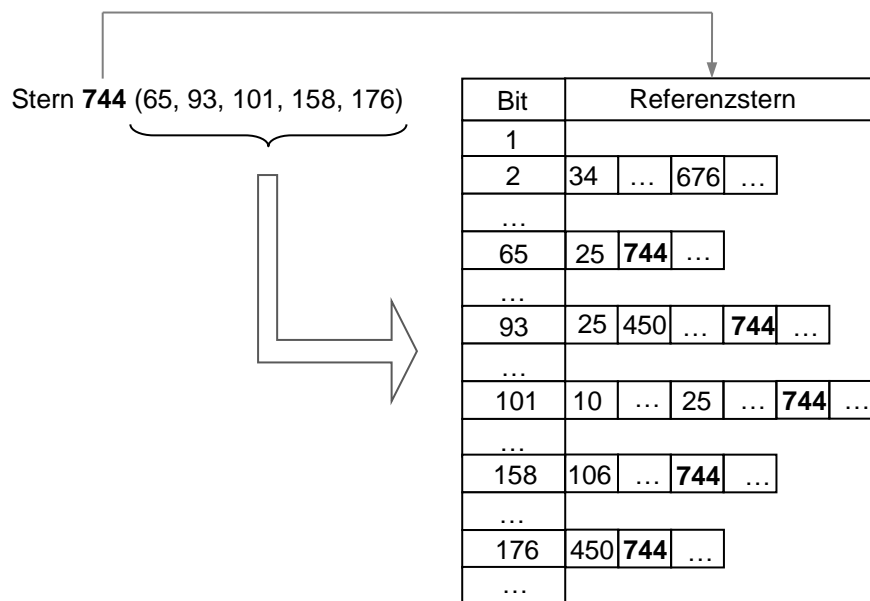


Abbildung 63. Aufbau der Datenbank für den Gitter-Algorithmus

Der Suchverfahren ist relativ schnell. Beim Durchsuchen der Datenbank werden nur die Zeilen betrachtet, welche im aus dem Bild ermittelten Mustervektor als nicht leer gekennzeichnet sind. Im abgebildeten Beispiel sind die Zeilen 65, 93, 101, 158 und 176 zu durchsuchen. Dabei werden alle Sterne in diesen Zeilen bewertet. Es wird zusammengezählt, wie häufig die einzelnen Sterne in diese Zeilen vorkommen (siehe Tabelle 11). Der Stern mit der größten Häufigkeit gewinnt, da es am besten mit dem gesuchten Referenzstern aus dem aufgenommenen Bild übereinstimmt.

Tabelle 11. Bewertung der Sterne durch Gitter-Algorithmus

Stern-IDs	10	25	34	450	676	744
Checksumme	1	3	1	2	1	5
						Identifiziert

Dank des kompakten Aufbaus benötigt die Datenbank wenig Speicher. Außerdem sind die beim Musteraufbau notwendigen mathematischen Operationen relativ einfach. Die anspruchsvollsten davon sind Sinus/Cosinus - Berechnungen bei der Drehung der Sterne (siehe Abbildung 62 b). Diese Berechnungen können durch Verwendung des polaren Koordinatensystems vereinfacht werden [LEE03, LUO15] und entfallen z.B. bei Anwendung der Kreismustermethode.

- Kreismustermethode

Die Kreismustermethode [SIL06] ist eine Sternidentifikationsmethode, die der Gitter-Methode sehr ähnlich ist. Wie zuvor wird ein kreisförmiger Bereich des FOVs mit dem Radius $R > R_{min}$ um einen Polstern in der Nähe der Bildmitte betrachtet (siehe Abbildung 64 a) und b)). Der untersuchte Bereich wird in n gleichmäßige Ringe aufgeteilt. Wenn sich mindestens ein Stern in einem Ring befindet, wird dem Ring der Wert 1 zugewiesen. Leere Ringe bekommen den Wert 0. Das entstandene Bitmuster (siehe Abbildung 64 c)) wird nun mit den Einträgen in der Datenbank verglichen.

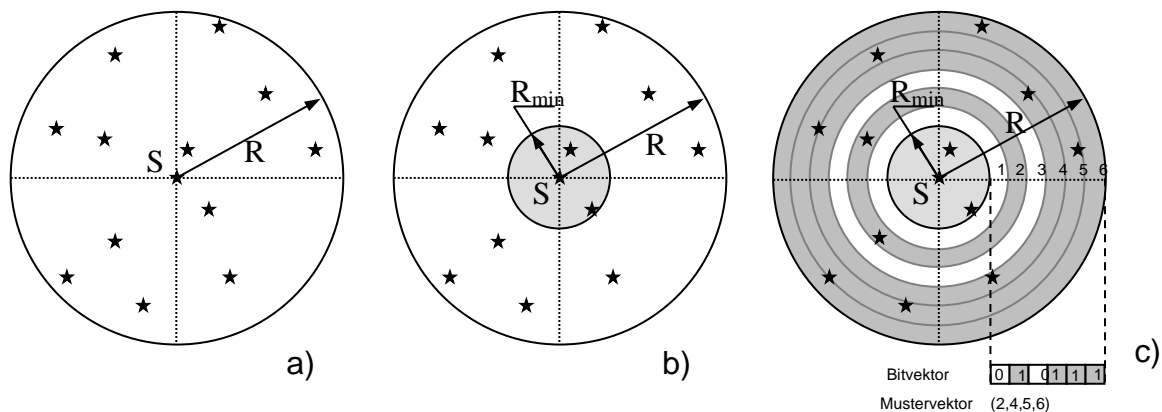


Abbildung 64. Kreismustermethode [SIL06]

Die Datenbank ist genauso aufgebaut wie beim Gitter-Algorithmus von Padgett. Auch das Identifikationsverfahren läuft nach dem gleichen Schema ab.

Weil das Kreismuster um einen Polstern gebildet wird und somit rotationssymmetrisch ist, wird die Berechnung vereinfacht der Schritt zur Drehung der Nachbarsterne entfällt. Dies bringt allerdings auch einen Nachteil mit sich: wenn sich mehrere Sterne in einem Ring befinden, werden ihre Positionen im Bild nur in einem Bit zusammengefasst. Das heißt, diese Sterne werden nicht berücksichtigt und gehen für die Identifikation praktisch verloren. Deshalb benötigt diese Methode mehr Sterne im Vergleich zur Gitter-Methode, um einen Stern eindeutig zu identifizieren.

- Identifikation durch ein neuronales Netz

Die ersten Versuche, ein neuronales Netz für eine Sternidentifikation zu nutzen, wurden bereits in den 90er Jahren unternommen. Je nachdem, welche Informationen für das Einlernen des neuronalen Netzes benutzt werden, können diese Sternidentifikationsmethoden als geometrische oder als Muster-Methode bezeichnet werden. Wenn für die Beschreibung einer Sternkonstellation geometrische Distanzen verwendet werden, so wird auch das neuronale Netz anhand der ansprechenden Geometrien trainiert [NA06, SPR09, LI03, ROB05, LIN97]. Andererseits können auch die kompletten Bilder ohne Vorverarbeitung an ein neuronales Netz übergeben werden. In diesem Fall übernehmen die Neuronen beide Aufgaben, Objekterkennung und Sternidentifikation. Die zeitintensive Suche in einer Datenbank entfällt. Allerdings sind das benötigte neuronale Netz und die damit assoziierten Berechnungen sowie der Lernprozess sehr umfangreich. Sie benötigen oft eine spezielle, energieintensive Hardware, um die Bearbeitung der Bilder zu beschleunigen. Aus diesem Grund sind sie aktuell noch nicht für kleine eingebettete Systeme geeignet.

5.5.7.8 Datenspeicherung und Suchmethoden

Eine der zeitintensivsten Operationen bei der Sternidentifikation ist das Durchsuchen der Datenbank nach einer geeigneten Sternkonstellation. Der Aufbau bzw. die Architektur der Datenbank ist von der verwendeten Identifikationsmethode abhängig und durch die verfügbare Hardware eingeschränkt (siehe Abbildung 65).

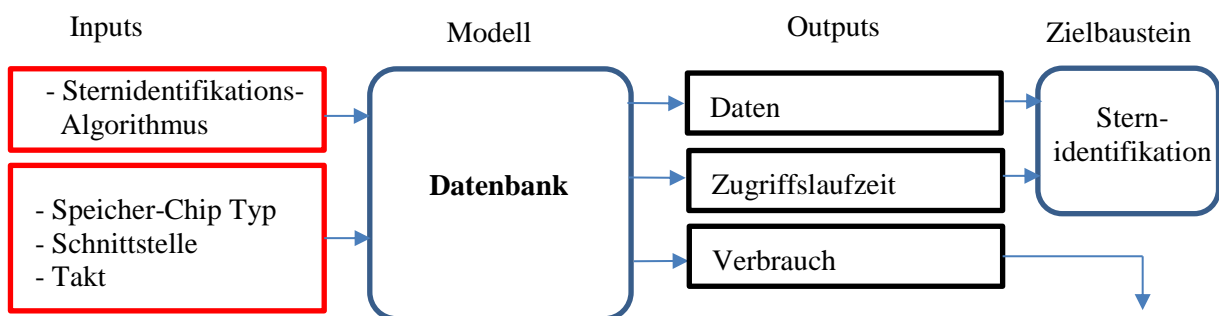


Abbildung 65. Datenbank-Modell

Bei der Sternidentifikation anhand einer der oben beschriebenen Dreieck- und Polygon-Algorithmen wird angestrebt, die geometrische Beschreibung des Sternbildmusters in der Datenbank so schnell wie möglich zu finden. Je schneller die Suche verläuft, desto zeiteffizienter ist die gesamte Lagebestimmung. Die Geschwindigkeit des gesamten Identifikationsverfahrens ist im Wesentlichen von der ausgewählten Suchmethode abhängig.

- Sequentielle Suche (Lineare Suchmethode)

Die lineare Suche ist die einfachste Methode, einen Zielwert in einer Datenbank bzw. Vergleichstabelle zu finden. Dabei werden die n Referenzwerte in der Datenbank nacheinander mit dem gesuchten Wert verglichen, bis der Eintrag, der dem gesuchten Wert entspricht bzw. diesem am nächsten ist, gefunden wurde.

Wenn der gesuchte Wert am Ende der Datenbank ist, muss die ganze Datenbank durchsucht werden, um diesen Wert zu finden. Die lineare Suchmethode benötigt relativ viel Zeit und ist

unter Umständen mit einer großen Anzahl an Speicherzugriffen verbunden. Sie hat eine Komplexität von $O(n)$, da sie im schlechtesten Fall n Vergleiche benötigt. Der Vorteil dieser Suchmethode besteht jedoch darin, dass sie für das Durchsuchen nicht sortierter Datenbanken geeignet ist.

- Divide-and-Conquer-Suche (Binäre Suche)

Die binäre Suchmethode erlaubt, das gesuchte Element in einem bestimmten Suchbereich einer Datenbank relativ schnell zu finden. Die Voraussetzung dafür ist, dass alle Einträge aufsteigend oder absteigend sortiert sind.

Bei dieser Methode wird zuerst das mittlere Element des Suchbereichs überprüft. Es gibt drei mögliche Ergebnisse des Vergleichs:

- Das mittlere Element ist größer als das gesuchte Element. Dann wird der ursprüngliche Suchbereich halbiert und nur der vordere Teil betrachtet.
- Das mittlere Element ist kleiner als das gesuchte Element. Dann betrachtet man nun nur die hintere Hälfte.
- Das mittlere Element ist unter Berücksichtigung der Messgenauigkeit gleich dem gesuchten Element. Der Suchvorgang ist erfolgreich beendet und das Zielelement ist gefunden.

Nachdem der Suchbereich halbiert wurde, wird das neue mittlere Element ermittelt und weiter nach gleichem Schema vorgegangen. Wenn der Suchbereich nur noch ein Element beinhaltet, wird die Suche beendet. Das letzte Element ist entweder das gesuchte Element oder es kommt im vordefinierten Suchbereich gar nicht vor. Diese Suchmethode gehört zur logarithmischen Komplexitätsklasse $O(\log n)$.

- Binärer Suchbaum

Auch diese Suchmethode verkürzt weiterhin die Suchzeit. Diese Methode wurde in [QUI96] für den LIS-Algorithmus verwendet.

Die Realisierung dieser Methode beginnt mit dem Aufbau eines binären Suchbaums (siehe Abbildung 66). Dies erfolgt offline, anhand der Daten von einem errechneten Vektor z.B. aus geometrischen Winkelabständen.

Als binärer Suchbaum wird ein geordneter Binärbaum bezeichnet, in dem

- a) jeder Knoten einen Wert hat, dessen Werte vergleichbar sind und die Baumstruktur bestimmen;

- b) für jeden Knoten, der kein Blatt ist, muss gelten, dass alle Knoten in seinem linken Teilbaum kleiner und alle im rechten Teilbaum größer sind als er selbst.

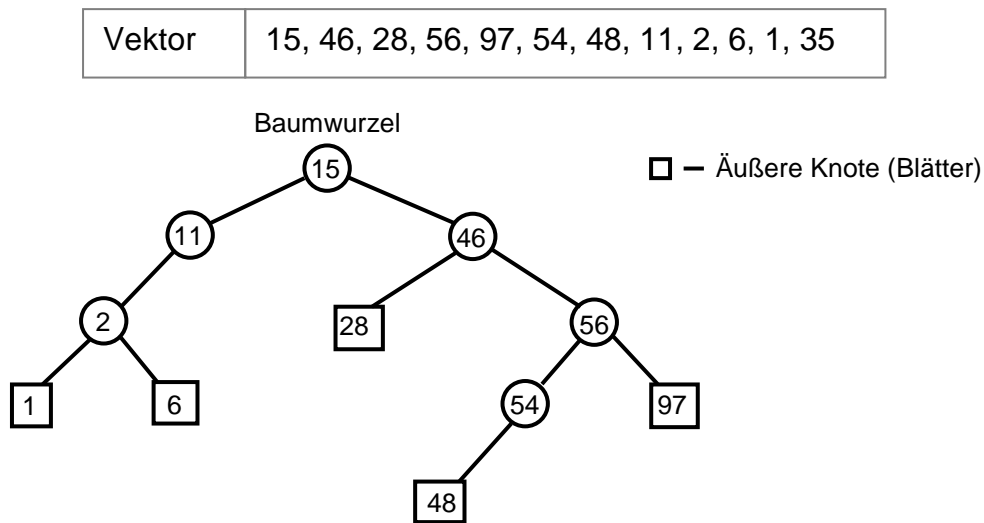


Abbildung 66. Binärer Suchbaum

Wenn der Suchbaum aufgebaut ist, wird er nicht flüchtig an Bord des Sternensensors gespeichert. Die Suche nach einem gewünschten Parameter beginnt von der Wurzel ausgehend an. Es wird überprüft, ob der jeweilige Wert im Knoten kleiner oder größer als der gesuchte Parameter ist. Wenn er kleiner ist, wird im linken Teilbaum weitergesucht, wenn er größer ist, im rechten. So wird bei jedem Suchschritt ein ganzer Teilbaum von der Weitersuche ausgeschlossen.

Die Suche ist abgeschlossen, wenn der gesuchte Wert in einem Knoten gefunden ist oder wenn der nächste zu überprüfende Teilbaum leer ist.

Der Suchaufwand ist stark von der Baumstruktur abhängig und proportional zur Baumhöhe h . Die Komplexität ist somit $O(h)$. Die Höhe des Suchbaums ist im Worst-Case so groß wie die Anzahl der vorhandenen Elemente n . Um einen optimalen und balancierten Aufbau erzielen zu können, ist es empfehlenswert, als Wurzel den Feldmittelwert zu definieren.

- K-Vektor

K-Vektor ist eine schnelle Suchtechnik, die von D. Mortari im Jahr 2000 für die Anwendung bei der Sternidentifikation vorgeschlagen wurde [MOR00].

In einem Suchfeld aus n Elementen, wird ein Element $y(i)$ im Genauigkeitsbereich $[y_a, y_b]$ gesucht. Zuerst wird das Feld y aufsteigend sortiert, daraus resultiert das Feld $s = y(I)$, mit $s(1) = y_{min}, s(n) = y_{max}$ und dem Vektor I mit Länge n , der die Sortieraufgabe beschreibt.

$$y = [0,35 \ 0,18 \ 0,72 \ 0,2 \ 0,6 \ 0,8 \ 0,51 \ 0,46 \ 0,62]$$

$$s = [0,18 \ 0,2 \ 0,35 \ 0,46 \ 0,51 \ 0,6 \ 0,62 \ 0,72 \ 0,8]$$

↓

Die Punkte $s(i)$ sind in Abbildung 67 dargestellt und werden mit einer linearen Funktion $z(i) = m \cdot i + q$ interpoliert.

Die Koeffizienten der linearen Funktion werden folgendermaßen berechnet:

$$m = \frac{y_{max} - y_{min} + 2\xi}{n-1}, q = y_{min} - m - \xi, \quad (97)$$

mit dem Berechnungsfehler ξ .

Für die Bestimmung des k -Vektors gilt:

- a) $k(1) = 0, k(n) = n;$
- b) $k(i) = j, \text{ mit } s(j) \leq z(i) < s(j+1) \text{ für alle } i \in [2, n-1].$

Praktisch gesehen ist $k(i)$ die Anzahl der $s(i)$ Punkte, die sich unter der linearen Funktion $z(i)$ befinden. Eine Beispielrechnung des k -Vektors ist in Abbildung 67 zu sehen.

$$k = [0 \ 2 \ 2 \ 3 \ 4 \ 5 \ 7 \ 8 \ 9]$$

Wenn der k -Vektor generiert ist, werden für den angegebenen Genauigkeitsbereich $[y_a, y_b]$ die Grenzen des Suchbereichs für den Vektor $s(j)$ ermittelt:

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \text{ und } j_t = \left\lceil \frac{y_b - q}{m} \right\rceil, \quad (98)$$

dabei bestimmen die Funktionen $\lfloor x \rfloor$ und $\lceil x \rceil$ die nächste Ganzzahl vor und nach dem Wert x . Für das Beispiel in Abbildung 67 ergeben sich $j_b = 5, j_t = 8$.

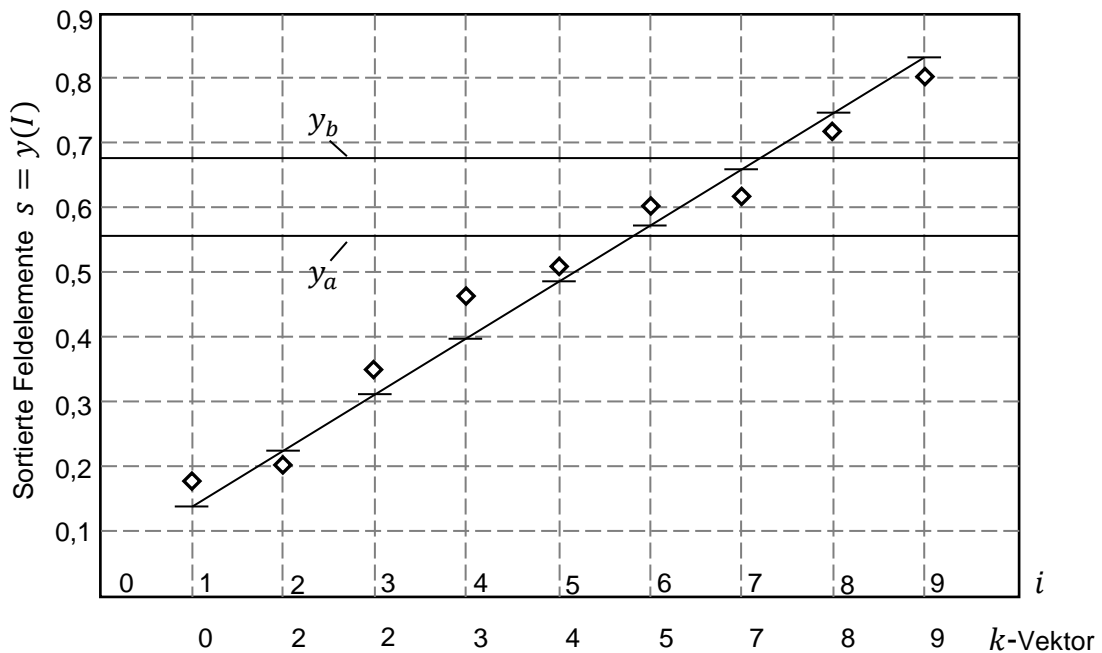


Abbildung 67. Berechnung des K -Vektors [MOR00]

Für die berechneten Grenzen können ganz leicht Start- und Endwert des k -Vektors definiert werden:

$$k_{start} = k(j_b) + 1 \text{ und } k_{end} = k(j_t)$$

Diese Werte erlauben eine direkte Identifikation der gesuchten Elemente $y(i) \in [y_a, y_b]$, dafür wird die inverse Rechnung $y(I(k))$ für den Bereich zwischen k_{start} und k_{end} durchgeführt.

5.5.7.9 Zusammenfassung der Sternidentifikationsmethoden

Im diesem Kapitel wurden lediglich diejenigen Identifikationsalgorithmen und Methoden betrachtet, die eine praktische Anwendung in Sternsensoren finden. Es existieren auch weitere Varianten und Kombinationen, die z.B. genetische Algorithmen, Singulärwertzerlegung (Singular Value Decomposition), Fuzzy-Logik oder Grafen-Theorie nutzen. Gute Zusammenfassungen und Vergleiche der verbreiteten Algorithmen zur Sternidentifikation bietet verschiedene Autoren [SPR09, PAD97-2, HO12, NA06, ZHA17].

Ein wichtiger Aspekt für eine eingebettete Implementierung ist die Komplexität der Algorithmen. Sie hat einen direkten Einfluss auf Hardwarearchitektur, Ausgaberate und Energieverbrauch. Die Komplexität einiger Algorithmen ist in Tabelle 12 dargestellt. Hier werden folgende Bezeichnungen verwendet: n - Anzahl der sichtbaren Sterne im Katalog, b - Anzahl der Sterne im Muster, f - Anzahl der Sterne im FOV, k - Länge des k -Vektors.

Tabelle 12. Komplexität der Algorithmen [SPR09]

Autor	Jahr	Mustercharakterisierung	Datenbank	Suchverfahren	Validierung
Liebe	1992	$O(f \lg b)$	$O(n)$	$O(n)$	
Quine	1996	$O(f \lg b)$	$O(n)$	$O(\lg n)$	
Paggett	1997	$O(f)$	$O(n)$	$O(n)$	
Mortari	1997	$O(b)$	$O(nf)$	$O(k)$	$O(bk^2)$
Kolomenkin	2008	$O(b)$	$O(nf)$	$O(k)$	$O(kf^2)$

Eine der effizientesten Methoden zur Sternidentifikation ist der Gitter-Algorithmus von Paggett. Die Datenbankgröße und die Komplexität dieses Verfahrens wachsen lediglich linear zu n . Die Berechnungen zur Mustercharakterisierung sind aber etwas aufwendiger im Vergleich zu geometrischen Methoden. Genau dieser Nachteil wurde durch Silani [SIL06] mittels Anwendung der Kreismustermethode anstatt eines Gitters weitestgehend eliminiert.

5.5.7.10 Lageberechnung

Die Lageberechnung ist der letzte Schritt der On-Board-Datenverarbeitung. Sie wird nach einer erfolgreichen Identifizierung von mindestens zwei Sternen durchgeführt. Für die identifizierten Sterne werden zunächst die Sterneinheitsvektoren im körperfesten Koordinatensystem des Sternsensors berechnet. Die entsprechenden Referenzvektoren sind für jeden Stern im Sternkatalog gespeichert, allerdings im International Celestial Reference Frame (ICRF). Die Kernaufgabe der Lageberechnung besteht darin, aus den Paaren von gemessenen Stern- und Katalogvektoren die relative Lage zu kalkulieren. Dies ist ein rein mathematisches Problem und die Geschwindigkeit der Berechnung ist lediglich von der ausgewählten Hardware abhängig (siehe Abbildung 68).

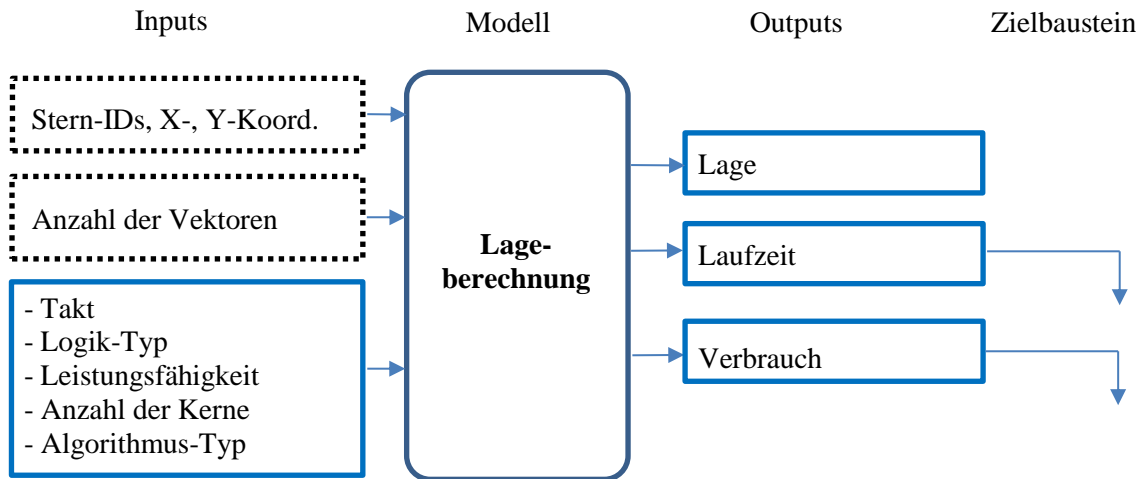


Abbildung 68. Modell der Lageberechnung

Die einfachste Methode, die Lage bezüglich eines Referenzkoordinatensystem zu ermitteln, ist der TRIAD-Algorithmus. Dieser Algorithmus benötigt zwei gemessene Vektoren \vec{v}_1, \vec{v}_2 im Sensorkoordinatensystem und zwei entsprechende Vektoren \vec{w}_1, \vec{w}_2 im Referenzkoordinatensystem. Anhand dieser Vektoren werden zwei orthogonale Koordinatensysteme r und t gebildet. Die ersten Achsen \vec{r}_1 und \vec{t}_1 beider Koordinatensysteme sind kollinear zum ersten Vektor \vec{v}_1 und \vec{w}_1 . Die zweiten Achsen \vec{r}_2 und \vec{t}_2 sind orthogonal zu den Ebenen, die durch die ersten Achsen \vec{r}_1 bzw. \vec{t}_1 und die zweiten Vektoren \vec{v}_2 bzw. \vec{w}_2 beschrieben sind. Die dritten Achsen \vec{r}_3 bzw. \vec{t}_3 sind dann jeweils orthogonal zu den ersten beiden Achsen. Wenn beide Koordinatensysteme berechnet wurden, kann die Lagematrix A durch einfache Matrizenmultiplikation bestimmt werden. Diese Operationen können mathematisch wie folgt beschrieben werden:

$$\begin{aligned} \vec{r}_1^e &= \vec{v}_1 / |\vec{v}_1| & \vec{t}_1^e &= \vec{w}_1 / |\vec{w}_1| \\ \vec{r}_2^e &= \vec{v}_1 \times \vec{v}_2 / |\vec{v}_1 \times \vec{v}_2| & \vec{t}_2^e &= \vec{w}_1 \times \vec{w}_2 / |\vec{w}_1 \times \vec{w}_2| \\ \vec{r}_3^e &= \vec{r}_2 \times \vec{r}_1 & \vec{t}_3^e &= \vec{t}_2 \times \vec{t}_1 \end{aligned} \quad (99)$$

$$[A] = [\vec{r}_1^e \quad \vec{r}_2^e \quad \vec{r}_3^e] \cdot [\vec{t}_1^e \quad \vec{t}_2^e \quad \vec{t}_3^e]^T \quad (100)$$

Die Lagematrix A kann anschließend in eine beliebige Darstellungsform, z.B. Quaternionen, umgewandelt werden.

Wenn mehr als zwei Vektorpaare, z.B. von drei oder mehr Sternen, für die Bestimmung der Lage zur Verfügung stehen, ist die Berechnung nicht eindeutig und überbestimmt. Diese Problematik ist als Wahba's Problem bekannt, das bereits in Jahr 1965 untersucht wurde [WAH65]. Um die optimale Lage aus mehr als zwei Vektorpaaren bestimmen zu können, wird die folgende Kostenfunktion minimiert:

$$J(A) = \frac{1}{2} \sum_{k=1}^n \omega_k |v_k - A \cdot w_k|^2. \quad (101)$$

Für jeden gemessenen Vektor v_i und den entsprechenden Vektor w_i im Referenzkoordinatensystem stellt ein Gewichtungsfaktor ω_i die Glaubwürdigkeit der Messung dar.

Für Wahba's Problem existieren zahlreiche Lösungsansätze, die zur Lagebestimmung beispielsweise die Methode der kleinsten Quadrate, die q-Methode, eine Quaternionenberechnung mit Kalman Filter oder die Singulärwertzerlegung verwenden [MAR88, MAR00, STE94, LIN02]. Eine der bekanntesten und in der Raumfahrt gut bewährten Methoden ist QUEST (Quaternion Estimator) [SHU81]. Diese Methode ist eine Erweiterung der q-Methode und erlaubt eine relativ genaue Abschätzung eines optimalen Quaternions aus den Vektorpaaren. Außerdem kann sie effektiv und schnell auf eingebetteten Systemen arbeiten [MAR93, MAR00, DUA09].

5.5.7.11 Zusammenfassung der Modellbildung

Im Unterkapitel 5.5 wurden verschiedenen Varianten und mathematische Beschreibungen der einzelnen Bestandteile eines Sternsensors vorgestellt. Je nach Systemanforderungen können nicht nur unterschiedliche Parameter, sondern auch verschiedene Methoden frei kombiniert werden, um eine maximale Performance zu erzielen. Allerdings sind nicht alle Methoden für eine eingebettete Implementierung in kleinen Sternsensoren gut geeignet. Bereits während der Durchführung des STELLA-Vorhaben wurden verschiedene Methoden und Algorithmen analysiert und bewertet. Die folgenden wurden für den STELLA-Sternsensor ausgewählt und implementiert:

- Sternerkennung: Sequentielle Regionen-Markierung
- Sternidentifikation: Kreismustermethode
- Lagebestimmung: TRIAD und QUEST.

Diese Methoden sind absolut voneinander unabhängig und daher modular einsetzbar. Sie benötigen keine spezielle Hardware und können sowohl auf einem rudimentären 8-Bit Mikrocontroller als auch auf einem High-End-FPGA laufen [FIS14]. Außerdem ist der Speicherbedarf relative niedrig und beträgt je nach Grenzmagnitude im Durchschnitt 2Mbyte. Deshalb bieten diese Methoden optimale Voraussetzungen für Anwendungen in Pico- und Nanosatelliten und sind bereits in Orbit verifiziert. Sie wurden auch für das AROS-Vorhaben verwendet und auf diverse Plattformen portiert.

5.6 Systemoptimierung

Fast jedes technologische Problem kann als Optimierungsproblem betrachtet werden. Systemingenieure sind täglich mit solchen Designproblemen unterschiedlicher Komplexität aus verschiedenen Bereichen konfrontiert. Durch gesammelte fachliche Erfahrungen, persönliche Kreativität und vorhandene CAD-Hilfstools erreicht ein Entwicklungsingenieur eine bestimmte Systemreife, die alle gestellten Anforderungen an das System erfüllt. Während der Lösungsfindung werden diverse Entscheidungen getroffen, Varianten gebildet, Systemparameter ausgewählt und letztendlich die Leistungsfähigkeit bewertet. All diese Prozesse in der Industrie und im Leben können als Optimierung bezeichnet werden.

Um während der Optimierung das Design variieren zu können, stehen dem Entwickler verschiedene Hebel in Form von Designvariablen zur Verfügung. Je nach Anzahl der Designva-

riablen und der Natur des Systems können Optimierungsprobleme unterschiedlich komplex in ihrer Beschreibung und Lösung sein. Sie haben aber alle eine ähnliche Struktur.

5.6.1 Problembeschreibung bei der Multiparameter-Optimierung

Die meisten Optimierungsprobleme beinhalten drei wesentliche Elemente. Diese beschreiben den Kontext des Problems in mathematischer Form [SIO17, GRI18, COL04].

Designvariablen bzw. Entscheidungsvariablen (Decision variables) stellen die quantitative Beschreibung der Entscheidungen, die bei der Optimierung des Systems getroffen werden sollen, dar. Ein Optimierungsproblem hat immer mindestens eine Entscheidungsvariable:

$$x_i \in \mathbb{R}, i = 1, \dots, n,$$

oder in Vektorform:

$$\vec{x} \in \mathbb{R}^n.$$

Die Kostenfunktion bzw. Zielfunktion (Objective function) bildet ab, wie gut die ausgewählten Entscheidungen sind. Abhängig vom Problemtyp wird versucht, eine Kostenfunktion $f(\vec{x})$ entweder zu minimieren oder zu maximieren:

$$\text{minimiere } f(\vec{x}) \text{ oder maximiere } f(\vec{x}).$$

Während des Optimierungsprozesses können Entscheidungsvariablen theoretisch beliebig geändert werden. In der Praxis formalisiert ein Optimierungsproblem reale Vorgänge, die aus physikalischen, technischen oder finanziellen Gründen limitiert sein können. Diese Limits werden **Randbedingungen oder Nebenbedingungen** (Constraints) genannt. Man unterscheidet zwei Typen von Randbedingungen:

$$\text{Ungleichheitsbedingungen: } \vec{g}(\vec{x}) \leq 0, \vec{g}(\vec{x}) \in \mathbb{R}^m,$$

$$\text{Gleichheitsbedingungen: } \vec{h}(\vec{x}) = 0, \vec{h}(\vec{x}) \in \mathbb{R}^p.$$

Außerdem können die Randbedingungen explizit oder implizit sein. Explizite Randbedingungen, z.B. das Projektbudget oder die maximalen Abmessungen des Produkts, werden im Optimierungsproblem genannt. Implizite Nebenbedingungen ergeben sich aus den physikalischen Beschränkungen der Designvariablen. So dürfen beispielweise geometrische Abstände nicht negativ sein. Zusammenfassend lässt sich sagen, dass Randbedingungen die Größe des sichtbaren Suchraums \mathbf{X} und des Kriterien-Raums \mathbf{Z} bestimmen:

$$\mathbf{X} \{ \vec{x} \mid \vec{g}(\vec{x}) \leq 0, \vec{h}(\vec{x}) = 0 \}$$

$$\mathbf{Z} \{ f(\vec{x}) \mid \vec{x} \in \mathbf{X} \}$$

In der Vektorform kann ein beliebiges Optimierungsproblem mit einer Zielfunktion wie folgt dargestellt werden:

$$\begin{aligned} & \text{minimiere } f(\vec{x}) \\ & \text{mit } \vec{g}(\vec{x}) \leq 0 \\ & \text{und } \vec{h}(\vec{x}) = 0 \end{aligned} \tag{102}$$

mit $\vec{x} \in \mathbb{R}^n$, $\vec{g}(\vec{x}) \in \mathbb{R}^m$, $\vec{h}(\vec{x}) \in \mathbb{R}^p$.

5.6.2 Klassifikation der Optimierungsmethoden

Bevor die spezifische Problematik der multikriteriellen Optimierung betrachtet wird, ist es sinnvoll, zuerst die Klassifikation und Besonderheiten der Algorithmen nur mit einem einzigen Optimierungsziel zu beginnen.

5.6.2.1 Methoden mit einem Optimierungsziel (Monozielmethoden)

An erster Stelle werden die Optimierungsmethoden in zwei Hauptgruppen unterteilt (siehe Abbildung 69): kombinatorische und kontinuierliche Methoden [COL04, BAN12].

Die kombinatorische Optimierung, auch diskrete Optimierung genannt, befasst sich mit der Suche nach einer optimalen Lösung aus einer endlichen Anzahl an Alternativen. Das Optimierungsproblem besteht aus einer großen Menge diskreter Elemente, z.B. den Strecken zwischen den Orten im Problem des Handlungsreisenden oder den Gegenständen im Rucksackproblem. Die optimale Lösung wird als eine Teilmenge dieser diskreten Elemente dargestellt. Dabei müssen die gestellten Nebenbedingungen erfüllt und die Kostenfunktion minimiert werden.

Kombinatorische Optimierungsverfahren werden weiterhin in exakte und in Approximationsmethoden unterteilt. Die exakten Methoden finden die korrekte Lage des globalen Optimums im Suchraum immer mit einer vorgegebenen Genauigkeit. Dies geschieht, indem der Suchraum Schritt für Schritt verkleinert wird. Ein typischer Vertreter dieser Klasse ist die „Branch-and-Bound“-Methode. Solche Methoden sind für Optimierungsprobleme mit sehr großen und komplexen Suchräumen jedoch nicht geeignet bzw. zeitlich ineffizient.

Approximationsmethoden werden für eine breite Klasse von Optimierungsproblemen verwendet, die in Polynomzeit nicht exakt gelöst werden können. Sie nutzen diverse Näherungsverfahren, um möglichst nah an das globale Optimum heranzukommen. Es wird zwischen heuristischen und metaheuristischen Approximationsmethoden unterscheiden. Heuristische Methoden wurden für spezifische Probleme entwickelt und nutzen das begrenzte Wissen über ein System, um allgemeingültige Aussagen über das Verhalten des Gesamtsystems zu treffen. Diese Methoden zeichnen sich durch geringe Laufzeit und geringen Rechenaufwand aus. Dazu gehört z.B. der Bergsteigeralgorithmus oder das Nelder-Mead-Verfahren.

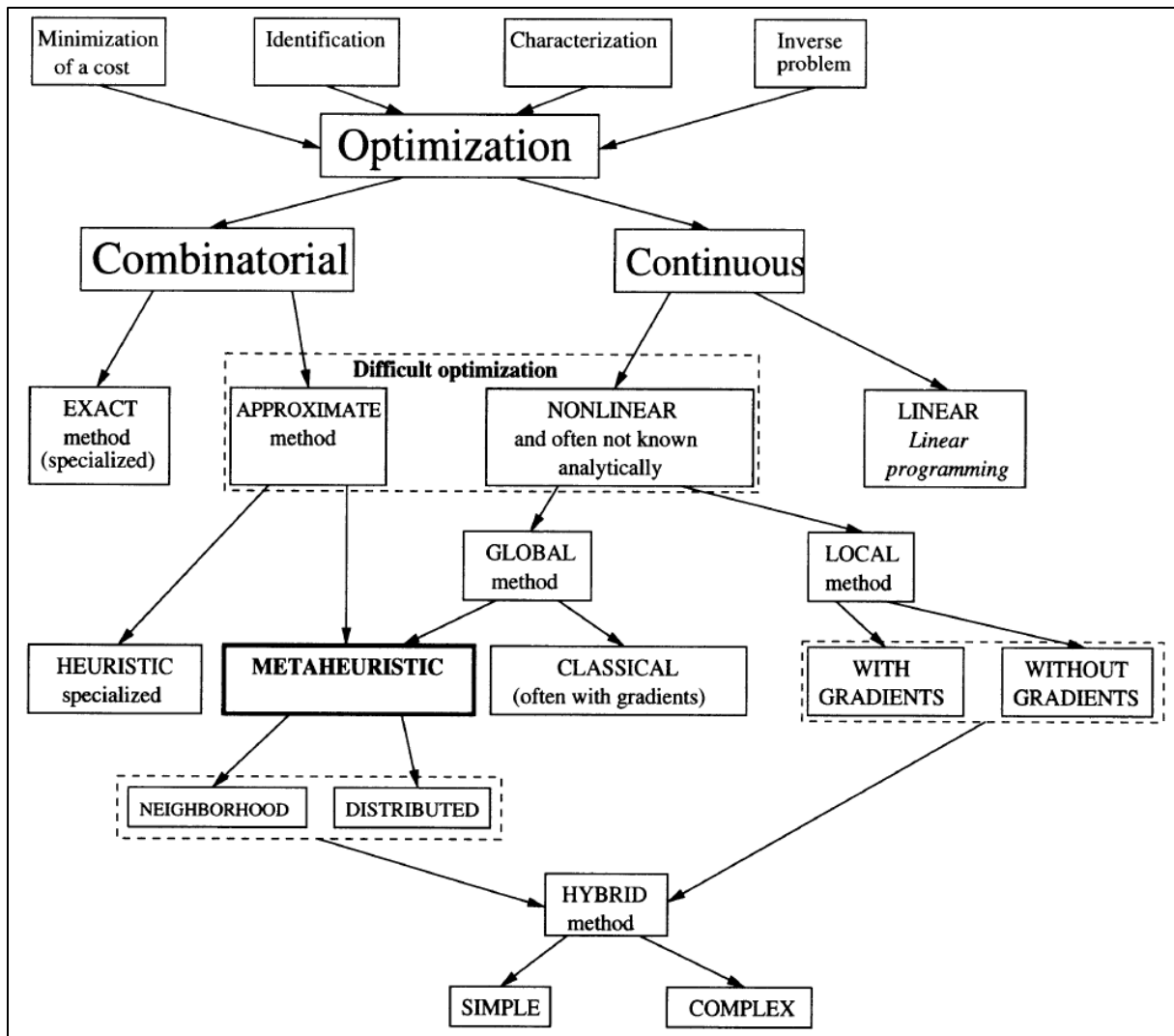


Abbildung 69. Klassifikation der Strategien mit einem Optimierungsziel [COL04]

Metaheuristische Methoden sind unabhängig vom Problemursprung und sind allgemeingültig. Sie beruhen größtenteils auf natürliche Prozesse der Physik, Biologie oder Ethologie. Wie auch in der Natur erlauben diese Methoden eine temporäre Verschlechterung der Systemleistungsfähigkeit. Dadurch können sie aus einem lokalen Optimum entkommen. Die bekanntesten metaheuristischen Methoden sind:

- Simulierte Abkühlung,
- Evolutionäre Algorithmen,
- Sintflutalgorithmus,
- Ameisenalgorithmus,
- Tabusuche und
- Partikelschwarmoptimierung.

Diese Methoden haben den Vorteil, dass sie relativ leicht implementierbar sind und für komplexe Optimierungsprobleme wesentlich schneller laufen als die exakten Methoden [BAN12]. Außerdem finden sie in der gegebenen Simulationszeit immer eine gültige Lösung. Allerdings ist diese Lösung oft suboptimal.

Ein zweiter Zweig der Optimierungstechniken befasst sich mit kontinuierlichen Problemen. Hier werden zwei Problemgruppen behandelt: lineare und nicht lineare. Die linearen Methoden beschränken sich auf diejenigen Probleme, die ausschließlich durch lineare Gleichungen und Ungleichungen beschrieben werden können [SIO17, GRI18]:

1. Entscheidungsvariablen: $\vec{x} \in \mathbb{R}^n$
2. Kostenfunktion: $\vec{f}(\vec{x}) = \vec{c}\vec{x}$, $\vec{c} \in \mathbb{R}^k$
3. Randbedingungen: $\vec{a}_e\vec{x} = \vec{b}_e$, $\vec{a}_g\vec{x} \geq \vec{b}_g$, $\vec{a}_l\vec{x} \leq \vec{b}_l$, mit $\vec{a}_e, \vec{b}_e \in \mathbb{R}^{m_e}$, $\vec{a}_g, \vec{b}_g \in \mathbb{R}^{m_g}$, $\vec{a}_l, \vec{b}_l \in \mathbb{R}^{m_l}$

Bei der nichtlinearen Optimierung enthalten Kostenfunktion und Nebenbedingungen nicht-lineare Komponenten, die analytisch oft nur schwer umgeformt werden können. Dazu gehören zwei Methodengruppen: globale und lokale. Lokale Methoden nutzen beispielweise Ableitungen der Zielfunktion, um das Optimum zu finden. Unter Verwendung dieser Technik ist es nicht immer möglich, das globale Optimum zu finden. In diesem Fall wird die Methode mit konstruktiven oder stochastischen Methoden kombiniert und erweitert, um auf die globale Ebene zu kommen. Dabei werden entweder Entscheidungsvariablen nacheinander modifiziert und die Modifizierung bereits geänderten Variablen verboten oder in der Nachbarschaft zufällig nach besseren Lösungen gesucht. Ein anderer Weg das globale Optimum eines nichtlinearen Problems zu erreichen, ist die Verwendung der Metaheuristiken als globale Suchmethode. Diese Methoden sind für die Lösung sowohl diskreter (kombinatorischer) und kontinuierlicher Probleme gut geeignet.

5.6.3 Auswahl des Optimierungsverfahrens zum Sternensensordesign

Um eine passende Optimierungsmethode zum Design von Sternensensoren selektieren zu können, müssen mehrere Auswahlkriterien analysiert werden. Die Kriterien sind in Tabelle 13 zusammengefasst und bieten eine gute Übersicht, um die richtige Methodenklasse auszuwählen.

Tabelle 13. Auswahlkriterien für die Optimierungsstrategie

Kriterien	Sternensensoranwendung
1. Anzahl der Entscheidungsvariablen bzw. Parameter: -eine -mehrere	Multivariable
2. Typ der Entscheidungsvariablen: -kontinuierlich -Ganzzahl oder diskret -Kombinatorisch	Kombinatorisch
3. Typ der Zielfunktionen bezüglich der Variablen -linear -quadratisch -nichtlinear	Nichtlinear
4. Anzahl der Zielfunktionen: -eine -mehrere	Multikriteriell
5. Form des Problems: -Mit Nebenbedingungen und Randbedingungen -Ohne Nebenbedingungen und Randbedingungen	Beschränkt durch Neben- und Randbedingungen

Die Designoptimierung von Sternsensoren ist sowohl ein kombinatorisches als auch nichtlineares Problem. Laut der Klassifikation in Kapitel 4.6.2. sind die metaheuristischen Methoden für die Lösung dieser beiden Probleme gut geeignet. Da die Metaheuristik relativ zeitintensiv ist, wäre eine verteilte Suche, die leicht parallelisierbar ist, vorteilhaft. Darüber hinaus sollte eine Optimierungsmethode ausgewählt werden, die mit einer großen Anzahl an Designvariablen umgehen kann. In diesem Fall bieten die evolutionären Algorithmen optimale Voraussetzungen.

Evolutionärer Algorithmen wurden in der Raumfahrt bereits erfolgreich zur Optimierung von Raumfahrzeugen und deren Komponenten eingesetzt [COM03, MOS98, HOR06]. Auch automatische Designoptimierungen von terrestrischen optischen Systemen wurden mithilfe evolutionärer Algorithmen durchgeführt [ONO00, LI10, THI05]. Deshalb wurde als primäre Methode zur Optimierung des Sternsensordesigns eine evolutionäre Strategie ausgewählt.

5.6.4 Evolutionsstrategie

In Laufe von Millionen von Jahren entwickelte die Natur eine perfekte Technik, um existierende biologische Arten zu stärken und deren Anpassungsfähigkeiten an veränderliche Umgebungen zu maximieren. Evolutionäre Algorithmen nehmen die natürliche Evolution als Vorbild für die Lösung von Optimierungsproblemen. Man simuliert eine Evolution, inklusive aller dazugehörigen Abläufe und Komponenten, wie Individuen, Population, Selektion, Mutation, Partnersuche und Rekombination. Zu diesen Algorithmen gehören genetische Algorithmen, Evolutionsstrategien, evolutionäre Programmierung und genetische Programmierung. Diese unterscheiden sich in der Darstellungsform der Individuen, der Operatoren und dem typischen Anwendungsbereich [NIS97, GRI18, GER04, WEI15].

5.6.4.1 Grundkonzept der Evolutionsstrategie

Evolutionsstrategien wurden in den 60er-Jahren entwickelt. Dieser Typ an Algorithmen ist besonders bequem für die Lösung von Problemen mit kontinuierlichen Entscheidungsvariablen und somit optimal geeignet für eine Optimierung des Sternsensordesigns. Mit etwas Anpassung können auch diskrete bzw. kombinatorische Variablen verarbeitet werden. Die Entscheidungsvariablen stellen die phänotypischen Eigenschaften des Individuums, in diesem Fall eines Sternsensors, dar.

- Darstellung des Individuums

Für die Lösung des Problems wird eine Funktion F angenommen, die zu optimieren ist. Diese Funktion ist abhängig von n Entscheidungsvariablen $x_i \in R, i = 1, 2 \dots n$. Alle Entscheidungsvariablen bilden zusammen einen Entscheidungsvektor bzw. Design-Vektor. Jedes Individuum wird mithilfe eines solchen Vektors kodiert. Außerdem benötigt jedes Individuum ein Maß der Streuung der Werte in Form der Standardabweichung σ_i oder einem gültigen Wertebereich $[r_i^-, r_i^+]$. Somit wird jedes Individuum als Vektor $a_j = (x_{ij}, \sigma_{ij})$ oder $a_j = (x_{ij}, r_i^-, r_i^+)$ dargestellt. Im weiteren Verlauf wird die erste Schreibweise verwendet.

- Initialisierung

In der ersten Phase wird eine Ausgangspopulation $P(0)$ von μ Individuen $a_j = (x_{ij}, \sigma_{ij})$, $j = 1, 2, \dots, \mu$ und $i = 1, 2, \dots, n$ initialisiert. Dabei sollen die Individuen möglichst gleichmäßig im Suchraum verteilt sein. In der Praxis ist es auch üblich, die Ausgangspopulation zufällig innerhalb der vorgegebenen Grenzen der Entscheidungsvariablen $[r_i^-, r_i^+]$ zu initialisieren. In [NIS97] wird es empfohlen, die Werte möglichst gleichmäßig zu verteilen, sodass die Anfangsabweichung $\sigma(0)_{ij}=3,0$.

- Bewertung der Ausgangspopulation

Jedes Individuum wird anhand seiner Fitness bewertet. Die Fitnessfunktion Φ ist identisch zur Zielfunktion F :

$$\Phi(a_j) = F(x_{ij}). \quad (103)$$

- Erzeugen der Nachkommen

In dieser Phase werden aus μ Eltern λ Nachkommen erzeugt. Es existieren diverse Methoden, wie die Nachkommen erzeugt werden können. Empfohlen wird ein Verhältnis μ/λ von ungefähr 1/7. So produzieren 15 Eltern üblicherweise etwa 100 Nachkommen [NIS97, GER04].

In diesem Abschnitt wird eine „vollständige“ evolutionäre Strategie verfolgt. Diese Art der Evolutionsstrategie beinhaltet nicht nur Mutationen, sondern auch einen Rekombinationsoperator zwischen den beiden Elternteilen. Sie ist besonders nah an der natürlichen Evolution der Säugetiere und kann in drei Schritte unterteilt werden.

1. Auswahl der Eltern

Eine stochastische Partnerwahl ist einer der einfachsten Grundformen für die Selektion der Eltern. Man bestimmt zufällig zwei Eltern E_1 und E_2 aus der aktuellen Population. Dabei haben alle Individuen die gleiche Wahrscheinlichkeit $1/\mu$, ausgewählt zu werden. Eine weitere mögliche Strategie zur Partnerwahl ist elitär. Hier werden nur die besten Eltern einen oder mehrere Nachkommen erzeugen.

2. Rekombination und Erzeugen der Nachkommen

Bei der Rekombination tauschen die Eltern ihr genetisches Material aus und erzeugen dadurch Nachkommen. Die Nachkommen werden im Suchraum um die Eltern zerstreut (siehe Abbildung 70). Man unterscheidet zwischen zwei Typen der Rekombination diskrete: *a*) und intermediäre *b*).

Bei der diskreten Rekombination wird ein Nachkommen durch einen direkten Austausch der Entscheidungsvariablen erzeugt. Für jede Variable wird entschieden, von welchem Elternteil ein Wert übernommen wird [POH99]:

$$x_{ij} = x_{iE_1} \cdot p_i + x_{iE_2} \cdot (1 - p_i), \quad (104)$$

mit $p_i \in \{0,1\}$. Der zufällig ausgewählte Wert p_i bestimmt die Wahrscheinlichkeit zur Übernahme der Variable von Elternteil x_{iE_1} .

Auf gleiche Art und Weise wird mit der Standardabweichung der Designvariablen umgegangen. Dies gilt auch für intermediäre Varianten der Rekombination. Bei dieser Rekombinationsart können die Entscheidungsvariablen der Nachkommen einen Zwischenwert annehmen:

$$x_{ij} = x_{iE_1} \cdot p_i + x_{iE_2} \cdot (1 - p_i), \quad (105)$$

wobei der Skalierungsfaktor $p_i \in [-d, 1 + d]$ für jede Variable neu bestimmt wird. Für d wird ein Wert zwischen 0,25 und 0,5 empfohlen. Der Wert d kann auch variiert werden, um die Dominanz des einen oder anderen Elternteils zu steuern.

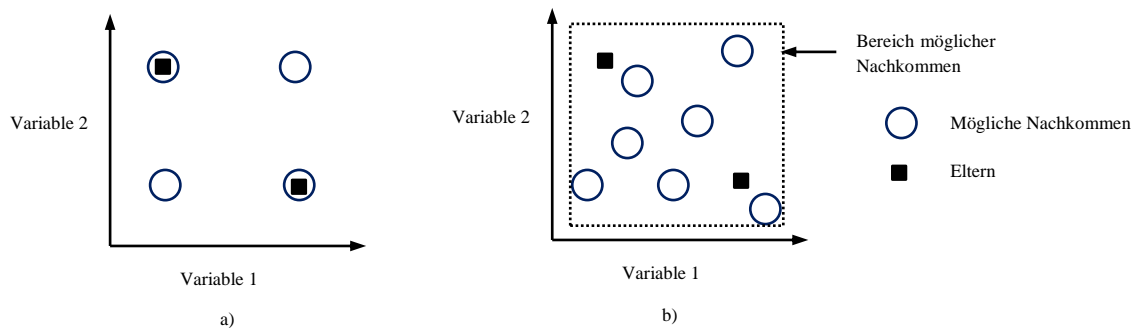


Abbildung 70. Diskrete und intermediäre Rekombination [POH99].

Wenn der Skalierungsfaktor dynamisch angepasst werden soll, muss sichergestellt werden, dass die Werte der Entscheidungsvariablen im gültigen Bereich $[r_i^-, r_i^+]$ liegen. Dies kann wie folgt geschehen [BEL99]:

$$x_{ij} = x_{iE_1} \cdot p_i + x_{iE_2} \cdot (1 - p_i), \quad (107)$$

mit $x_{iE_1} \geq x_{iE_2}$ und $p_i \in [-d_i^-, 1 + d_i^+]$ uniformverteilt in diesem Bereich:

$$d_i^- = \frac{x_{iE_2} - r_i^-}{r_i^+ - r_i^-} \quad (106)$$

$$d_i^+ = \frac{r_i^+ - x_{iE_1}}{r_i^+ - r_i^-}$$

3. Mutation der Nachkommen

In diesem Schritt werden die Nachkommen mutiert. Die Mutation sorgt für eine zufällige Diversität in der Population. Somit bleiben alle Bereiche des Suchraums erreichbar. Zur Einfachheit der Erläuterung wird angenommen, dass alle Entscheidungsvariablen die gleiche Standardabweichung haben. Die Standardabweichung wird in jedem Evolutionsschritt für jedes Individuum erneut berechnet [SCH94, NIS97]:

$$\sigma_{ij}^{neu} = \sigma_{ij} \cdot e^{(\tau_1 \cdot N(0,1) + \tau_2 \cdot N_{ij}(0,1))}. \quad (108)$$

Hier steuern Koeffizienten τ_1 und τ_2 die Schrittweite der Mutation und können frei gewählt werden. Empfohlen sind Werte zwischen 0,1 und 0,2. Die erste standardverteilte Zufallszahl $N(0,1)$ wird nur einmalig am Anfang der Evolutionsstrategie erzeugt und beeinflusst die Änderung aller Standardabweichungen. Die zweite standardverteilte Zufallszahl $N_{ij}(0,1)$ wird individuell in jedem Evolutionsschritt j für jede Variable i neu generiert.

Die neuen Entscheidungsvariablen erhalten anhängig von der Zufallszahl $N'_{ij}(0,1)$ folgende Werte:

$$x_{ij}^{neu} = x_{ij} + \sigma_{ij}^{neu} \cdot N'_{ij}(0,1). \quad (109)$$

In manchen Fällen ist es ratsam, die Werte der Entscheidungsvariablen während der Mutation in bestimmten Grenzen zu halten. Dafür wird eine andere Form der Mutation erforderlich [BEL99, POH99]:

$$x_{ij}^{neu} = x_{ij} + sign \cdot range_i \cdot \delta, \quad (110)$$

mit $sign \in \{-1,1\}$, gewählt mit gleicher Wahrscheinlichkeit, dem Mutationsbereich $range_i = \rho(r_i^+ - r_i^-)$, $\rho \in [0.1, 0.5]$ und

$$\delta = 2^{-k\beta}, \quad (111)$$

mit $\beta \in [0, 1]$, gleichverteilt in diesem Intervall und $k \in N^+$, der Präzision der Mutation. Die letzte ist analog zur Schrittweite der Mutation.

Der größte Teil der mutierten Nachkommen wird in der Nähe der nicht mutierten Individuen generiert. Sowohl bei der ersten als auch bei der zweiten Mutationsform werden kleine Änderungen bevorzugt. Durch die Verwendung der Exponentialfunktion bzw. Gaußverteilung werden kleine Änderungen entlang der Achsen wahrscheinlicher als große. Diesen Effekt verdeutlicht die nachfolgende Abbildung 71.

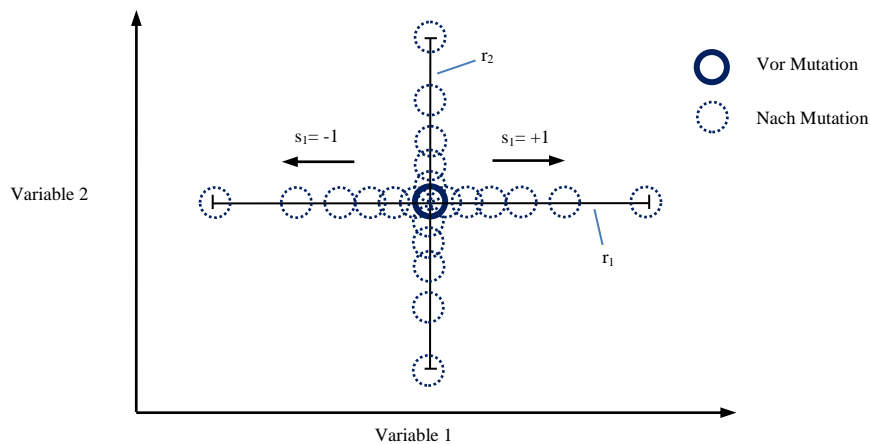


Abbildung 71. Auswirkung des Mutationsoperators [POH99]

- Bewertung der Nachkommen und Erzeugen der Zwischenpopulation

Jedes neue Individuum wird anhand seiner Fitness bewertet:

$$\Phi(a_l^{neu}) = F(x_{il}^{neu}), \quad l = 1, 2, \dots, \mu. \quad (112)$$

und zur anfangs leeren Zwischenpopulation hinzugefügt.

- Selektion

In diesem Schritt wird die neue Population gebildet. Die Auswahl der Individuen erfolgt z.B. nach dem Elite-Prinzip. Nur die fittesten schaffen es in die nächste Runde. Man unterscheidet zwischen zwei Typen der Selektion: „Plus“ und „Komma“.

Bei der Plus-Selektion werden die μ besten Nachkommen aus der Gemeinschaft der Eltern- und Zwischenpopulation ausgewählt. Deshalb nennt man diese Strategie $(\mu + \lambda)$. Der Vorteil dieser Strategie besteht darin, dass die beste gefundene Lösung immer in die nachfolgende Generation übernommen wird. Der Nachteil ist, dass diese Variante sehr schnell lokal konvergieren kann, da die besten Nachkommen aus vergangenen Evolutionsschritten früh anfangen zu dominieren.

Bei der Komma-Selektion werden die besten Individuen nur aus dem μ Nachkommen der Zwischenpopulation gebildet. Die Eltern-Generation wird bei dieser Variante nicht berücksichtigt und stirbt aus. Die Gefahr der vorzeitigen Konvergenz in einem lokalen Optimum ist bei der Komma-Strategie (μ, λ) geringer. Die Komma-Selektion bietet jedoch keinerlei Garantie, dass die beste Lösung in die nächste Generation übernommen wird. Deshalb wird empfohlen, die beste Lösung separat zwischenspeichern.

- Terminierung der Evolution

Bei der Suche nach einem globalen Minimum wird über mehreren Generationen iterativ vorgefahren, bis einer der Abbruchkriterien die Evolution unterbricht. Dies geschieht, wenn die gewünschte Anzahl an Generationen überschritten wird oder die Individuen zu einer Lösung

konvergieren, d.h. die Individuen mit der besten und der schlechtesten Fitness unterscheiden sich kaum voneinander:

$$|\Phi_{worst} - \Phi_{best}| \leq C, \quad (113)$$

mit erforderlicher Genauigkeit C bei der Suche nach optimaler Lösung.

Nach dem Abbruch der Evolution stellt das beste Individuum die Lösung des gestellten Optimierungsproblems dar.

5.6.5 Multikriterielle Optimierungsmethoden

Bis jetzt wurden nur Probleme mit einem Optimierungsziel betrachtet. Wenn mehr als eine Funktion gleichzeitig optimiert werden sollen, spricht man von multikriterieller Optimierung oder Mehrzieloptimierung. Jedes neue Optimierungsziel fügt dem System einen zusätzlichen Freiheitsgrad hinzu und verändert bzw. erweitert den Suchraum:

$$\begin{aligned} & \text{minimiere } \vec{f}(\vec{x}) \\ & \text{mit } \vec{g}(\vec{x}) \leq 0 \\ & \text{und } \vec{h}(\vec{x}) = 0 \end{aligned} \quad (114)$$

mit $\vec{x} \in \mathbb{R}^n$, $\vec{f}(\vec{x}) \in \mathbb{R}^k$, $\vec{g}(\vec{x}) \in \mathbb{R}^m$, $\vec{h}(\vec{x}) \in \mathbb{R}^p$. In der Beschreibung gibt es nun k Zielfunktionen, die gleichzeitig optimiert werden sollen.

Um eine Lösung der multikriteriellen Optimierung anschaulich darstellen zu können, sollen ein paar neue Begriffe eingeführt werden. Im Gegensatz zur Monozieloptimierung generieren multikriterielle Optimierungsmethoden nicht nur eine Lösung, sondern eine Menge an Lösungen. Diese Lösungen werden als Pareto-Lösungen und die entsprechenden Werte der Zielfunktionen als Pareto-Front bezeichnet (siehe Abbildung 72).

Als Pareto-optimal wird eine Lösung $\vec{x}^* \in \mathbf{X}$ bezeichnet, wenn keine andere Lösung $\vec{x} \in \mathbf{X}$ existiert, für welche gilt $\vec{f}(\vec{x}) \leq \vec{f}(\vec{x}^*)$ und $f_i(\vec{x}) < f_i(\vec{x}^*)$ für mindestens ein Kriterium, $i = 1, \dots, k$. In diesem Fall sagt man auch, dass die Lösung \vec{x}^* dominiert.

Oft liefern multikriterielle Optimierungsalgorithmen keine Pareto-optimalen Lösungen. Die gelieferten Lösungen erfüllen jedoch die gestellten Anforderungen, sind technisch realisierbar und somit bedeutsam für die praktische Anwendung. Diese Lösungen werden als schwache Pareto-Optima bezeichnet. Für ein schwaches Pareto-Optimum $\vec{x}^* \in \mathbf{X}$ gilt, dass es keine andere Lösung $\vec{x} \in \mathbf{X}$ gibt, für die $\vec{f}(\vec{x}) < \vec{f}(\vec{x}^*)$.

Die multikriteriellen Optimierungsmethoden werden oft in drei Kategorien unterteilt: priori, progressiv und posteriori. Diese definieren, wann die Abwägung bezüglich der Zielfunktionen durchgeführt wird.

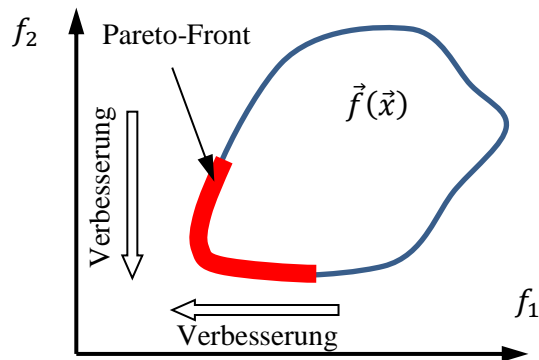


Abbildung 72. Pareto-Front für eine zweidimensionale Zielfunktion eines Minimierungsproblems [NGA05]

Die klassischen Methoden, die sowohl für eine priore als auch eine posteriore Auswertung der Lösungen geeignet sind, versuchen, ein multikriterielles Problem anhand spezieller Vektoroperatoren in einen Satz von Problemen mit nur einem Optimierungskriterium umzuwandeln [NGA05, MAR04, KON06, COL04]:

- Methode der gewichteten Summe oder des gewichteten Produkts

Eine der einfachsten Methoden, eine Kostenfunktion aus mehreren Zielfunktionen zu bilden, ist die gewichtete Summe:

$$\text{minimiere } \Phi = \sum_{j=1}^k w_j \cdot f_j(\vec{x}), \quad (115)$$

mit den Gewichtungskoeffizienten w_j . Diese werden wie folgt gewählt: $w_i \geq 0$, $\sum_{j=1}^k w_i = 1$. Diese Methode ist sehr leicht zu implementieren, aber die Ergebnisse sind nur so gut wie die Wahl der Gewichtungskoeffizienten und oft nicht Pareto-optimal. Die Auswahl der Gewichte kann ohne Kenntnisse über das Systemverhalten problematisch werden. Außerdem ist die Erreichbarkeit der Lösungen im nicht konvexen Bereich eines Pareto-Frontest äußerst erschwert bzw. nicht gegeben.

Wenn die Optimierungskriterien Werte in verschiedenen Größenordnungen liefern, ist es nicht möglich, passende Gewichtungskoeffizienten für die gewichtete Summe zu wählen. In diesem Fall ist die Verwendung der Methode des gewichteten Produkts von Vorteil:

$$\text{minimiere } \Phi = \prod_{j=1}^k (f_j(\vec{x}))^{w_j} \quad (116)$$

Die Berechnung des gewichteten Produkts ist allerdings nicht linear und unter Umständen etwas zeitintensiver.

- Exponentielles gewichtetes Kriterium

Eine weitere Variation nutzt eine exponentielle gewichtete Funktion und kann mit nicht-konvexen Pareto-Fronten umgehen:

$$\text{minimiere } \Phi = \sum_{j=1}^k (e^{p \cdot w_j} - 1) e^{p \cdot f_j(\vec{x})}. \quad (117)$$

Je nach gewählter Potenz p können jedoch relativ große Werte entstehen und zum Überlaufen führen.

- Lexikographische oder hierarchische Methode

Bei dieser iterativen Methode werden die einzelnen Kriterien bzw. Zielfunktionen nach Priorität sortiert und schrittweise einzeln minimiert:

$$\text{minimiere } f_i(\vec{x}), \quad (118)$$

mit $x \in X$ und $f_j(\vec{x}) \leq \left(1 + \frac{\delta_j}{100}\right) f_j(\vec{x}_j^*)$, $j = 1, 2, \dots, i - 1$ für $i = 1, 2, \dots, k$. Hier ist i die Position einer Zielfunktion in der Hierarchie und j ist die Iterationsnummer. In jeder Iteration wird ein Optimum $f_j(\vec{x}_j^*)$ bestimmt. Die Variable δ_j kann im Bereich von 0 bis 100 verändert werden und verschärft die gestellte Nebenbedingung.

- Methode der begrenzten Zielfunktionen, ε – Nebenbedingung

Diese Methode reduziert die multikriterielle Optimierung auf die wichtigste Zielfunktion $f_s(\vec{x})$. Alle anderen Kriterien werden zu den Nebenbedingungen umgewandelt.

$$\text{minimiere } f_s(\vec{x}),$$

$$\text{wobei } l_i \leq f_i(\vec{x}) \leq \varepsilon_i \quad (119)$$

$$\text{mit } \vec{g}(\vec{x}) \leq 0$$

$$\text{und } \vec{h}(\vec{x}) = 0$$

mit $\vec{x} \in X$, $i = 1, 2, \dots, k$, $i \neq s$. Diese Methode existiert in diversen Variationen, so kann z.B. bei der ε – Nebenbedingung die untere Grenze l_i entfallen. Auch diese Methode führt nicht unbedingt zu einer Pareto-optimalen Lösung. Eine systematische Änderung der Grenze ε_i verbessert die Situation, jedoch kann dies dazu führen, dass die Lösung außerhalb des sichtbaren Bereiches X liegt.

- Methode der Zielprogrammierung

Für jedes Optimierungskriterium $f_j(\vec{x})$ definiert der Entscheidungsträger einen Zielwert t_j . Nur wird nicht mehr die reine Zielfunktion, sondern die Abweichung vom Zielwert minimiert:

$$\text{minimiere } \Phi = \sum_{j=1}^k w_j |f_j(\vec{x}) - t_j|. \quad (120)$$

Außerdem wird jede Entscheidung mit einem eigenen Koeffizient w_j gewichtet. Die Auswahl der Gewichtungskoeffizienten und der Zielwerte verlangt Vorabinformationen über das System. Wenn $t_j = 0$, verwandelt sich die Zielprogrammierung in die Methode der gewichteten Summe.

Die oben genannten Methoden sind für beliebige Typen von Optimierungsalgorithmen gültig. Es existieren aber spezialisierten Methoden, die beispielsweise für evolutionäre Algorithmen bzw. genetische Algorithmen entwickelt wurden. Fünf solcher Methoden werden hier kurz behandelt [MAR04, NGA05, COL04, KON06]:

- *Vector Evaluted Genetic Algorithm (VEGA)*

Wie die klassischen Methoden versucht der VEGA-Algorithmus, das multikriterielle Problem auf ein Monoproblem zu reduzieren. Dies geschieht, indem die Gesamtpopulation P und k in Subpopulationen unterteilt wird. Jede Subpopulation der Größe μ/k wird nur anhand einer Zielfunktion $f_j(\vec{x}), j = 1, \dots, k$ evaluiert und dann stochastisch oder proportional zur Fitness selektiert. Die Operatoren für Mutation und Rekombination werden wie üblich ausgeführt. Am Ende wird die neue Population aus den Ergebnissen der einzelnen Subpopulationen zusammengestellt.

Die VEGA-Methode ist zwar leicht zu implementieren, führt jedoch dazu, dass die Lösungen nicht homogen über die Pareto-Front verteilt sind und Cluster bilden, die nur optimal für einzelne Kriterien sind. Die nachfolgenden Pareto-basierten Methoden versuchen, diesen Nachteil zu eliminieren:

- *Multi Objektive Genetic Algorithm (MOGA)*

Um Konvergenz der Optimierungsalgorithmen in den einzelnen Regionen der Pareto-Front zu vermeiden, werden verschiedene Nischen-Techniken im Kriterien- oder Entscheidungsraum angewendet. Als Nische wird ein Bereich im Kriterienraum bezeichnet, in dem sich die Lösungen sehr nah versammeln. Diese Techniken divergieren die Verteilung der Lösungen und unterscheiden sich im Wesentlichen nur durch das eingesetzte Verfahren zur Bestimmung des Fitnesswertes eines Individuums. So wird beim MOGA-Algorithmus z.B. der Fitnesswert eines Individuums, abhängig von der Anzahl der Nachbarn sowie der Anzahl der Individuen, über welche es dominiert, verändert.

- *Niched Pareto Genetic Algorithm (NPGA)*

Um Konvergenz in Richtung eines Kriteriums zu vermeiden, werden bei Selektion in NPGA zufällig ausgewählte Lösungspaare, auch Kandidaten genannt, einer Vergleichsgruppe gegenübergestellt. Die Vergleichsgruppe wird zufällig initialisiert. Die Größe der Gruppe ist variabel und beträgt etwa 10% der Gesamtpopulation. Nur die Kandidaten, die über die Ver-

gleichsgruppe dominieren, wirken bei der Erzeugung der nächsten Generation mit. Wenn beide Kandidaten nicht dominieren, wird ein sogenanntes Fitness-Sharing durchgeführt. Beim Sharing wird der Fitnesswert eines Individuums proportional zur Anzahl der Nachbarn in der gleichen Nische reduziert.

Diese Methode ist relativ schnell, da bei der Abwicklung von Selektion der Vergleichsoperationen nur zusammen mit der Vergleichsgruppe und nicht mit der ganzen Population ausgeführt werden.

- Non-dominated Sorting Genetic Algorithm (NSGA)

Der NSGA-Algorithmus nutzt eine Schichtklassifikation der Individuen. Die erste Schicht bilden die dominanten Lösungen \vec{x}_1 der Anfangspopulation P_1 . Alle diese Lösungen bekommen den gleichen und höchsten Fitnesswert. In der zweiten Iteration werden die Lösungen \vec{x}_1 aus der Population ausgeschlossen und eine neue Teilpopulation P_2 gebildet. Auch hier wird nach den dominanten Lösungen \vec{x}_2 für die zweite Schicht gesucht. Diese Lösungen erhalten einen etwas niedrigeren Fitnesswert. Diese Schritte werden solange wiederholt, bis die Population leer ist. In jeder Iteration wird zusätzlich Sharing als Nischen-Technik angewendet.

Die Selektion erfolgt wie üblich proportional zum Fitnesswert. Die Individuen in der ersten Schicht haben höhere Fitnesswerte und erzeugen somit mehr Nachkommen.

- Strength Pareto Evolutionary Algorithm (SPEA)

Beim SPEA-Verfahren werden die dominanten Lösungen in dünn besetzten Regionen durch höhere Fitnesswerte belohnt. Dies geschieht anhand einer externen Liste fester Größe N^* , wo die dominanten Lösungen P^* aus jeder Iteration gespeichert werden. Jede Lösung aus P^* bekommt einen Stärkegrad zugewiesen. Der Stärkegrad $s_i \in [0, 1)$ des Individuums $i \in P^*$ ist als relative Anzahl an Individuen aus P , über welche Individuum i dominiert, bezogen auf die Populationsgröße $N + 1$, definiert. Die Fitness der Individuen in der externen Liste beträgt $\Phi_i = s_i$.

Für ein Individuum j aus P wird die Fitness allerdings als Summe der Stärkegrade von über j dominanten Lösungen aus P^* plus Eins berechnet:

$$\Phi_j = 1 + \sum_i s_i, \quad (121)$$

mit $i \succ j$ (Individuum i dominiert über j) und $\Phi_j \in [1, N)$. Kleinere Fitnesswerte bedeuten bessere Reproduktionschancen. Diese Bewertungstechnik führt automatisch dazu, dass die Lösungen in der Nähe der Pareto-Front priorisiert werden. Diese verteilen sich zusätzlich über die Front, um bessere Fitnesswerte zu bekommen.

Zusammenfassend lässt sich sagen, dass die speziellen Methoden etwas komplizierter in der Umsetzung im Vergleich zu den klassischen Methoden sind und tiefere Kenntnisse des Systemverhaltens voraussetzen. Außerdem beeinflussen diese unter Umständen die Struktur der klassischen Evolutionsstrategie. Aus diesem Grund wurde in der ersten Version des Optimie-

rungstools vorerst nur die einfachste Methode der gewichteten Summe für das Sternsensorde-
sign umgesetzt.

6 Implementierung des Optimierungstools für die automatische Entwicklung von Sternsensoren

Dieses Kapitel befasst sich mit der konkreten Softwareimplementierung des Optimierungstools. Dies beinhaltet die typischen Schritte wie Implementierung der benötigten Funktionen, Initialisierung des Gesamtsystems und Validierung der Funktionsfähigkeit.

6.1 Implementierung des Tools

Die einzelnen Softwarebausteine des Optimierungstools wurden als MatLab-Skript im M-File-Format realisiert. MatLab wird oft für eine schnelle Lösung von wissenschaftlichen Problemen verwendet. Es verfügt über vorgefertigte Visualisierungsbausteine und ein dynamisches Speichermanagement. Außerdem ähnelt die MatLab-Skriptsprache C/C++ und kann somit bei Bedarf leicht portiert werden.

Die Implementierung ist angelehnt an die in Kapitel 5.4. vorgestellte Struktur. Die einzelnen Komponenten des Modells wurden nach den Vorgaben des Kapitels 5.5 umgesetzt. Um die Implementierung der Komponenten zu erleichtern, wurden diverse Annahmen getroffen bzw. eine reduzierte mathematische Beschreibung verwendet (siehe Abbildung 73):

- Für die Simulation der Abbildung der Sterne wurden drei Wellenlängen im sichtbaren Spektrum, 380 nm, 600 nm und 780 nm, benutzt.
- Bei der Ermittlung des Auflösungsvermögens wurde nur die theoretische Größe der Beugungsscheibchen ermittelt und diese mit der Spot-Größe verglichen. Keine weiteren Effekte der Lichtbeugung wurden untersucht.
- Durch fehlende Angaben in den Datenblättern kann kein vollständiges Modell der Bildsensoren erstellt werden. Stattdessen wurde eine Referenzaufnahme einer bekannten Sternkonstellation unter vorgegebenen Bedingungen erstellt und für die Modellierung verwendet.
- Bei der Optikmodellierung (paraxial und nichtparaxial) wurden nur die geometrische Spot-Form und dessen Lage beurteilt. Die Energieverteilung im Spot wurde nicht berücksichtigt.
- Verschmierung der Sterne durch die Bewegung des Satelliten wird nicht simuliert.
- Als Referenz wurde für Sternerkennung die sequentielle Regionenmarkierung, für Sternidentifikation die Kreismustermethode, und für Lagebestimmung der QUEST-Algorithmus verwendet.
- Die Laufzeiten der Sternerkennung und des QUEST- Algorithmus wurden als konstant angenommen und werden nicht simuliert.
- Um die Anzahl der Sterne in Sichtfeld beurteilen zu können, wurde der simulierte Sternhimmel mit einer festen Schrittweite abgescannt. Insgesamt wurden für jeden Sternsensor 216 gleiche Himmelsausschnitte untersucht.
- Die Modellierung der Bild- und Datenverarbeitung beschränkt sich nur auf die Ermittlung der Sternkatalog- bzw. Datenbankgröße und der durchschnittlichen Anzahl der benötigten Speicherzugriffe.

- Elektrische Hardware ist vom Sensorinterface abhängig und soll bereits zum Beginn der Modellierung ausgewählt werden.

Der Optimierungsteil des Tools wurde angelehnt an die Evolutionsstrategie aus Kapitel 5.6.4 umgesetzt. Es wurden zwei Rekombinationstypen, zufällig und elitär, implementiert. Außerdem kann zwischen „Koma“- oder „Plus“-Selektionsstrategie gewählt werden.

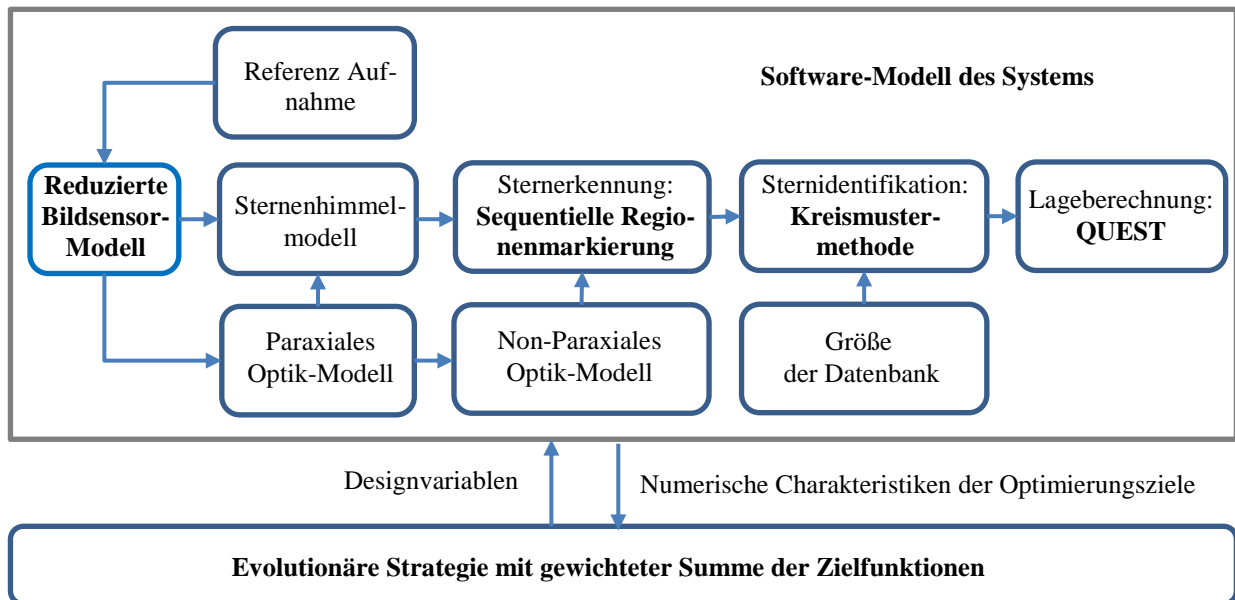


Abbildung 73. Struktur des reduzierten Optimierungstools

Die hierarchische Programmstruktur ist in Abbildung 74 dargestellt. Alle implementierten Funktionen mit dazugehörigem Datenfluss sind nach Ebenen geordnet. Die Ebene 0 entspricht *main()*. Das Optimierungstool hat kein Graphical User Interface (GUI) und wird direkt per Command Line gesteuert bzw. die Ausführung darin kontrolliert.

6.2 Parametrierung und Initialisierung des Tools

Leistungsfähigkeit und Laufzeit der Optimierung ist von den gewählten Parametern und den vorgenommenen Einstellungen abhängig. Es gibt etwa 60 verschiedene Größen, die vor dem Start des Optimierungstools initialisiert werden müssen. Die wichtigsten davon sind in Tabelle 14 zusammengefasst.

Besonders bedeutend für die Evolutionsstrategie ist die Größe der Population. Es wurde mit verschiedenen Populationen von 25 bis 500 Individuen experimentiert. Die Anzahl der Nachkommen wurde wahlweise auf 7 oder 15 gesetzt. Die Terminierung der Optimierung erfolgt nach einer vorgegebenen Anzahl an Generationen. Die Generationenanzahl wurde von 25 bis 400 variiert. Auch das Maß der Mutation ist einstellbar und sorgt für unverzichtbare Diversität zwischen den Nachkommen. Außerdem kann zwischen zwei verschiedenen Rekombinations- und Selektionsstrategien gewählt werden.

Jeder Sternsensor wird als Individuum in Form eines Vektors mit Designvariablen dargestellt. Die meisten dieser Variablen gehören zum Optikdesign. An erste Stelle wird die gewünschte Anzahl an Linsen festgelegt. Davon ausgehend werden alle optischen Flächen, das Linsenma-

terial und alle benötigten Abstände definiert. Am Anfang wird die Population zufällig, jedoch innerhalb vorgegebener Grenzen, initialisiert. Die Grenzen werden durch die Anforderungen und Randbedingungen an das Sternensensorsystem festgelegt. Diese filtern technisch nicht realisierbare oder für die Anwendung nicht geeignete Lösungen heraus. Solche Lösungen werden entweder bereits bei der Erzeugung komplett verworfen oder die Bewertung des dazugehörigen Individuums wird reduziert, d.h. die Individuen werden bestraft.

Für die Bewertung der Individuen wurden neun Zielfunktionen definiert (Tabelle 14). Diese werden anhand einer gewichteten Summe in einer Fitnessfunktion zusammengeführt. Zuvor werden die einzelnen Zielfunktionen skaliert bzw. normiert. Die Gewichtungsfaktoren können je nach Priorität vom Nutzer definiert werden.

6.3 Verifikation der Software

Bevor mit den Tests des gesamten Optimierungstools begonnen werden kann, wurde eine Plausibilitätsprüfung der einzelnen Funktionsbausteine durchgeführt. Vor allem die lineare und nichtlineare Optiksimation wurde gründlich untersucht, da hier die meisten Optimierungen stattfinden. Dafür wurden verschiedene bekannte Linsensysteme berechnet und die Ergebnisse mit den Originalquellen verglichen. Ein Beispiel der paraxialen Berechnung eines Dupletts ist in Tabelle 15 und Abbildung 75 dargestellt. Bei der paraxialen Berechnung wurden Bezeichnungen gemäß Kapitel 5.5.3 verwendet. Drei Strahlen wurden durch das System geführt: ein Hauptstrahl (rot), ein axialer Strahl (blau) und ein paralleler Strahl (grün). Die Angaben zum optischen System wurden größtenteils aus Beispiel D in [SMI00, Seite 35] entnommen. Die Ergebnisse der linearen Simulation stimmen mit dem Referenzbeispiel überein.

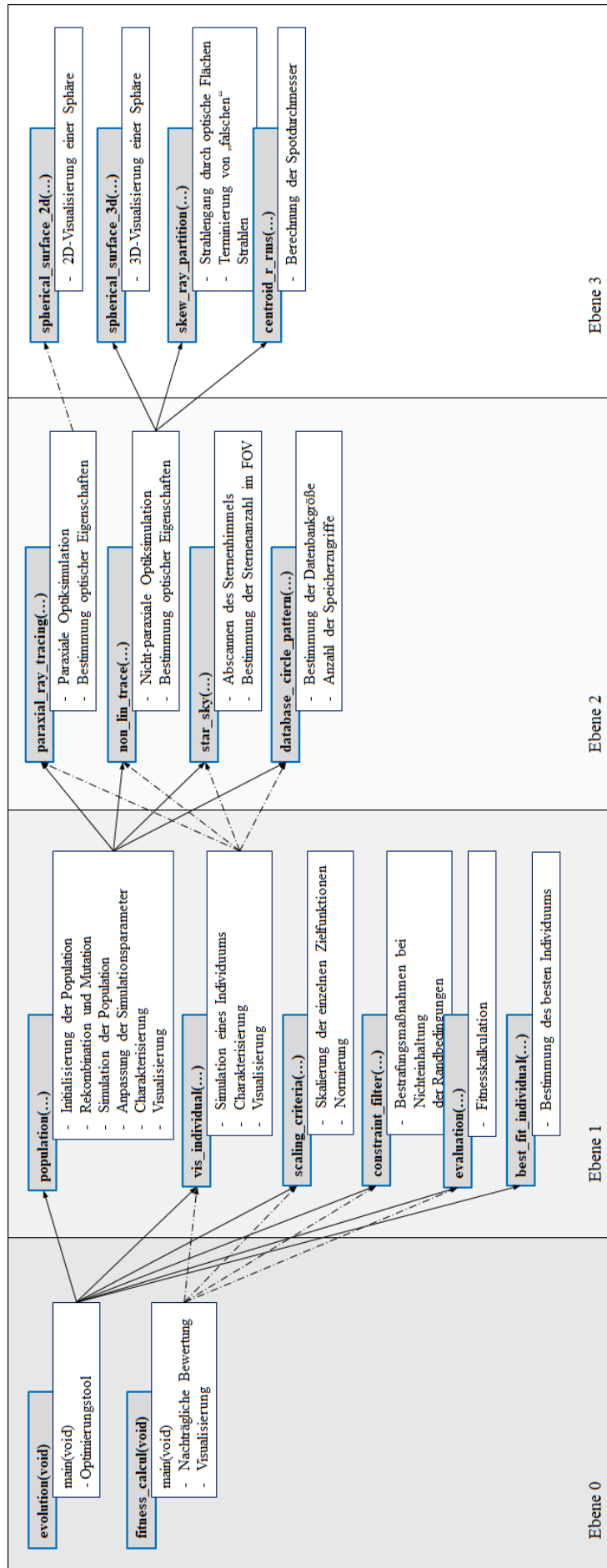


Abbildung 74. Hierarchische Struktur des Optimierungstools und der Hilfswerkzeuge

Systemteil	Inputparameter	Designvariablen	Randbedingungen	Output	Zielfunktion
Optik: paraxial_ray_tracing(...) non_lin_trace(...)	j – Anzahl der Linsen, oder s – Anzahl der Flächen (s=j+2)	t_i – Abstände Zwischen der Flächen, mit $i = 1, \dots, (s - 1)$ c_i – Krümmung (Flächenradien), mit $i = 1, \dots, s$ D_i – Flächendurchmesser, mit $i = 1, \dots, s$ NR_i – Glassorten, mit $i = 1, \dots, (s - 1)$	– Max. Objektvlänge – Auflagermaß – Max. Objektdurchmesser – Glaskatalog	– Durchmesser der Eintrittspupille (Öffnung) – Brennweite – Spot-Größe – Spot-Symmetrie – Verzeichnung – Vignettierung	– Spot-Größe – Spot-Symmetrie – Verzeichnung – Vignettierung
Detektor: population(...)	– Pixelgröße – Auflösung – Initiale – Grenzmagnitude – Brennweite – Öffnung	– Belichtungszeit	– Minimale Ausgaberate des Sternsensors – Grenzmagnitude	– Grenzmagnitude – Sichtfeld	
Datenbank: database_circle_pattern(...)	– Grenzmagnitude – Verzeichnung – Pixelgröße – Brennweite – Min. Ring			– Anzahl der Ringe – Datenbankgröße	– Anzahl der Ringe – Datenbankgröße – Anzahl der Speicherzugriffe
Sternhimmel: star_sky(...)	– Grenzmagnitude – Sichtfeld			– Max. Anzahl der Sterne – Min. Anzahl der Sterne – Ø-Anzahl der Sterne im Bild	– Min. Anzahl der Sterne – Ø-Anzahl der Sterne im Bild
Evolutionstrategie evolution(void) population(...)	μ – Populationsgröße λ – Anzahl der Nachkommen σ – Standardabweichungen der Mutation $\tau_{1,2}$ – Schrittweiten der Mutation		– Anzahl der Generationen	– Fitness aller Individuen – Bestes Individuum	

Tabelle 14. Parameter, Variablen und Zielfunktionen des Optimierungstools

Tabelle 15. Berechnungen des Strahlenverlaufs im Paraxialgebiet eines Dupletts.

t_k [mm]	300	0	10	2	199,68
n_k	1,0	1,0	1,5	1,6	1,0
c_k [mm ⁻¹]	0	0	0,02	-0,02	0
r_k [mm]	-	10	20	20	20

Außerdem wurden die wichtigen Systemgrößen, wie z.B. effektive Brennweite, Eintrittspupillenradius und Eintrittspupillenposition bestimmt (siehe Tabelle 16).

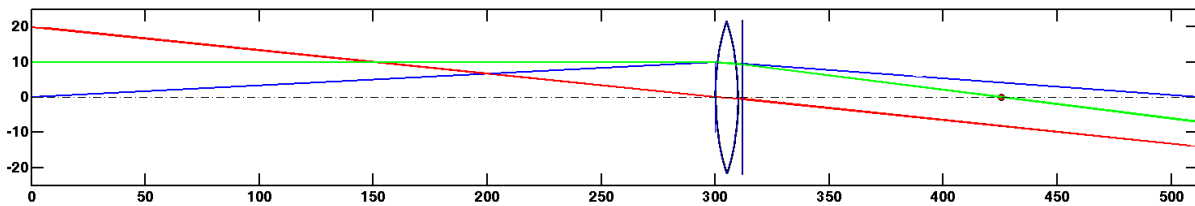


Abbildung 75. Paraxialer Strahlengang durch ein Duplett

Eine vergleichbare Rechnung wurde auch für ein Cooke-Triplet mittels des kommerziellen CAD-Tools WinLens3D sowie mit dem Optimierungstool durchgeführt. Die Eingabewerte für die Berechnung einer Folge der sphärischen Flächen kann Abbildung 76 entnommen werden. Wie deutlich zu sehen ist, stimmen auch hier die Ergebnisse der linearen Optikberechnung weitestgehend überein (siehe Tabelle 17). Die Abweichungen sind auf Rundungsfehler bei der Berechnung in MatLab und in WinLens3D zurückzuführen.

Tabelle 16. Errechneten Parameter des Dupletts

Parameter	Wert
Anzahl der Oberflächen	4
Aperturblende	2
Apertur [mm]	10
Feldblende	5
Effektive Brennweite [mm]	122,95
Schnittweite (back focal length) [mm]	113,50
Eintrittspupillenradius [mm]	10
Eintrittspupillenposition*	0
Austrittspupillenradius [mm]	9,87
Austrittspupillenposition** [mm]	-7,83
Durchmesser der Eintrittsluke [mm]	20,26
Position der Eintrittsluke* [mm]	-8,59
Durchmesser der Austrittsluke [mm]	20
Position der Austrittsluke** [mm]	0
Sichtfeld	136,73

*Relativ zur ersten Fläche, ** relativ zur letzten Fläche

	#	Srf	Film	Film wave	Radius	Seprn	Glass	Maker	n	V	Ap/2	Ap
0		Obj				Infinity	AIR					
1	1	S			37,4000	5,90	N-SK4	Schott	1,61	20,75	14,60	fixed
2		S			-341,4800	12,93	air				14,60	fixed
3	2	S			-42,6500	2,50	SF2	Schott	1,65	11,34	10,80	fixed
4		S			36,4000	2,00	air				10,40	fixed
5	3	*S			Plane	9,85	air				6,00	
6	4	S			204,5200	5,90	N-SK4	Schott	1,61	20,75	11,60	fixed
7		S			-37,0500	77,42	air				11,60	fixed
8		Img			Plane	-0,26	mm - def				6,19	

Abbildung 76. Eingabefeld in WinLens3D mit Beispieldaten eines Cooke-Triplets

Im nächsten Schritt wurde ein nicht paraxialer Gang der Strahlen durch die Linsen simuliert (siehe Abbildung 77). Hier ist die Verteilung der einzelnen Strahlen im Spot wichtig (siehe Abbildung 78).

Der Objektstand wurde auf unendlich gestellt. Dies entspricht den Aufnahmen von Sternen am realen Himmel. Es wurden drei Lichtwellenlängen, 380 nm, 600 nm und 780 nm, untersucht. Bei der Visualisierung mit dem Optimierungstool werden zwar alle drei Wellenlängen in gleichen Farbe abgebildet, dafür werden Spot-Mittelpunkt und RMS-Durchmesser für die einzelnen Wellenlängen dargestellt.

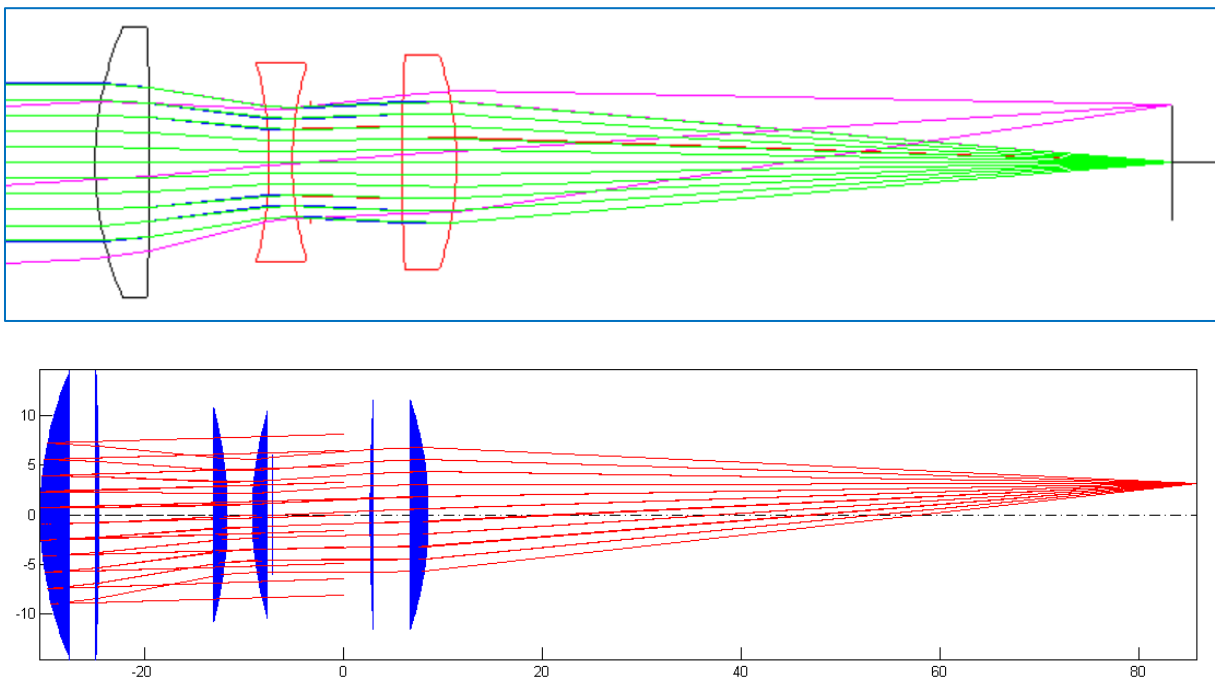


Abbildung 77. Strahlengang durch das Cooke-Triplet in WinLens3D (oben) und dem Optimierungstool (unten)

Wie bei der paraxialen Berechnung unterscheiden sich die Ergebnisse der Spotberechnung unwesentlich und liegen im Genauigkeitsbereich der Berechnungsmethoden.

Auch weitere Funktionen zur Sternenhimmelsimulation und zur Datenbankgenerierung wurden in mehreren Versuchen während des AROS Vorhabens validiert. Die Ergebnisse wurden mit echten Sternenaufnahmen verglichen, indem die Abstände zwischen den bestimmten Sternen kalkuliert wurden. Ein Beispiel eines solchen Vergleichs wurde bereits in

Abbildung 26 und Abbildung 27 illustriert.

Tabelle 17. Vergleich des Modells mit einem kommerziellen Tool Winlens 3D

Systemparameter	Ergebnisse der Modellierung	
	WinLens3D	OP Tool
Anzahl der Oberflächen	8	8
Nummer der Apertur-Fläche	5	5
Radius der Blende [mm]	6	6
Effektive Brennweite (paraxial) [mm]	101,191	101,1691
Schnittweite (back focal length) [mm]	77,423	77,4004
Eintrittspupillendurchmesser [mm]		17,0072
Eintrittspupillenposition * [mm]	30,519	30,5193
Austrittspupillendurchmesser [mm]		16,019
Austrittspupillenposition** [mm]	-17,890	-17,8905
Primäre Wellenlänge [nm]	600	600

*Bezogen auf die erste Fläche, **Bezogen auf Bildfläche

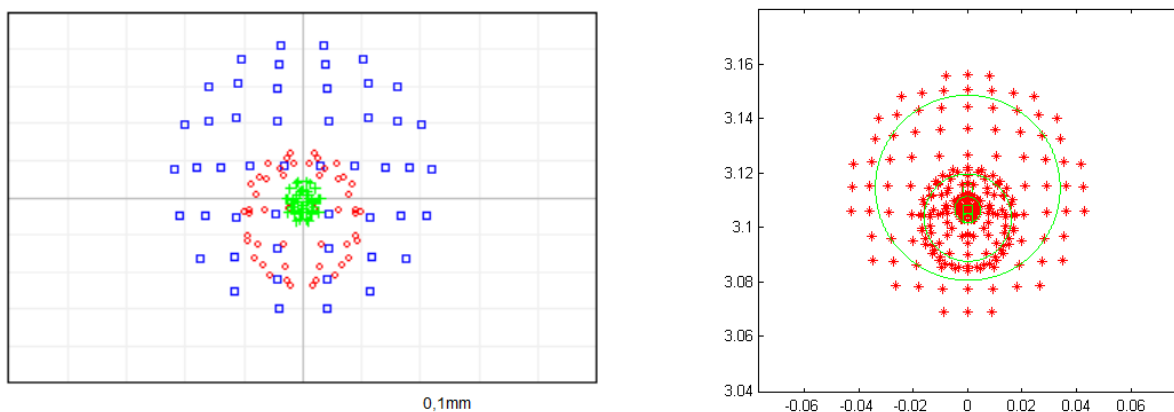


Abbildung 78. Spotbild von WinLens3D (links) und vom Optimierungstool (rechts)

6.4 Parallelverarbeitung

Das Optimierungstool wurde so aufgebaut, dass eine leichte Parallelisierung der Prozesse möglich ist. Vor allem ist eine gleichzeitige Berechnung von einzelnen Individuen bei der Evolutionsstrategie besonders zeiteffizient. In der ersten Implementierungsphase des Tools wurde aber auf eine Parallelisierung verzichtet und der Schwerpunkt der Entwicklung auf die Funktionalität gelegt, um die Grundidee des automatischen Designablaufs vollständig testen zu können. Die Optimierung des Sternensensordesigns lief dementsprechend relativ langsam. Für eine Population von 500 Individuen über 200 Generationen dauerten die Berechnungen etwa 7 bis 10 Tage. Die Berechnungen wurden auf einem Desktop-Rechner mit Intel® Core™ i7-4770 CPU @ 3,4 GHz und 8 GB RAM durchgeführt.

In der zweiten Phase wurde die Effizienz der MatLab Parallel Computing Tool Box untersucht. Als Hardware wurde eine Workstation mit zwei Intel® Xeon® E5-2640v2 mit 32 GB RAM verwendet. Jeder Prozessor verfügt über 8 Kerne und kann 16 Threads mit einer nominalen Taktfrequenz von 2,0 GHz verarbeiten. Daher wurden 32 Worker als virtuelle Ser-

verwendung initialisiert: 31 davon sind für die Optimierungsaufgaben zuständig und einer erledigt die Aufgaben des Schedulers und des Betriebssystems. Der nicht optimierte Programmcode wurde mit einem „parfor“-Operator, der für die Parallelisierung der Schleifen zuständig ist, bei der Berechnung einzelner Individuen versehen. Eine solche nicht zu feine Verteilung der Aufgaben minimiert den Informationsaustausch zwischen Worker und Scheduler. Der Code lief etwa 2,8-mal schneller im Vergleich zum oben genannten nicht parallelem Fall.

In der dritten Phase wurde der Programmcode weitgehend für die Anwendung des „parfor“-Operators optimiert und langsame Speicherzugriffe auf der Festplatte durch Preallocation reduziert. Durch diese Maßnahmen wurde eine etwa 5,6-fache Beschleunigung im Vergleich zum Code aus erster Phase erzielt.

7 Anwendung des Tools bei der Entwicklung von Sternsensoren für Pico- und Nanosatelliten

Für die Verifikation der Ergebnisse des automatischen Optimierungstools sollten zwei Designs generiert werden, die die Anforderungen der Pico- und Nanosatelliten-Missionen bezüglich Stromverbrauch, Abmessungen, Ausgaberate und Genauigkeit erfüllen. Anschließend wurden die beiden Designs vollständig ausgearbeitet, gebaut und qualifiziert. Auch die Verifikation im Orbit hat begonnen. Außerdem werden intensive Gespräche mit möglichen Partnern zur Kommerzialisierung und Weiterentwicklung der Prototypen geführt.

Bereits während der Entwicklung des STELLA-Sternsensors wurde viel Erfahrung im Bereich autonomer Sternsensoren gesammelt. Es wurden potentielle Stellen für eine weitere Verbesserung der Hardware und der On-Board-Software erkannt und bei der Neuentwicklung berücksichtigt. Außerdem wurden neue Technologien für Detektoren und Mikroelektronik untersucht und verwendet.

7.1 Ziele der Entwicklung und der Optimierung

Um Entwicklungsziele, Anforderungen und Randbedingungen genau festlegen zu können, wurden im Rahmen des AROS-Vorhabens zwei separate Fragebögen erstellt und an potenzielle Anwender im universitären Umfeld verteilt. Der erste Fragebogen bezog sich auf die Anwendung von Sternsensoren in Satelliten im CubeSat-Format bis zu 12U. Als Referenz wurde ein 3U-System vorgeschlagen. Der zweite Fragebogen bezog sich auf Nanosatelliten schwerer als 20 kg. Es wurden folgende Aspekte abgefragt:

1. Bei **funktionale Anforderungen und Randbedingungen** sollten Angaben z.B. zur Genauigkeit, Ausgabefrequenz, zulässigen Drehraten, Lebensdauer und operationellem Temperaturbereich gemacht werden.
2. Die **physikalischen Anforderungen und Randbedingungen** befassen sich mit Abmessungen, Masse, Formfaktor und Materialien der Sternsensoren.
3. **Elektrische Anforderungen und Randbedingungen** spezifizieren Energieverbrauch, benötigte Spannungsversorgung und elektrische Schnittstellen.
4. Zu **optischen Anforderungen und Randbedingungen** gehören FOV, Sonnen- und Erdausschlusswinkel. Letztere spezifizieren, unter welchen Bedingungen ein Sternsensor geblendet wird.
5. Die **operationellen Anforderungen und Randbedingungen** beinhalten betriebsrelevante Informationen wie benötigte Betriebsmodi, Angaben zum Redundanzkonzept, Möglichkeit des Katalog- und Softwareupdates sowie Bilddownloads.
6. Auch die **wirtschaftlichen Aspekte** bezüglich des vorstellbaren Preises und benötigter Menge wurden abgefragt, um die zukünftige Marktentwicklung prognostizieren zu können.

7. Angaben zu **zukünftigen Missionen** sollen konkrete Anwendungsfälle erläutern, um die Anforderungen und Randbedingungen besser verstehen zu können. Hier sind die Angaben zu ADCS und Nutzlast essenziell.

Die für die Optimierung relevanten Antworten wurden für zwei folgende Fallstudien Optimierung 1 und Optimierung 2 als Anforderungen und Randbedingungen aufbereitet.

7.1.1 Optimierung 1 auf die Bedürfnisse von Picosatelliten (OP1)

Bei den meisten Picosatelliten handelt es sich um CubeSats, von denen die meisten in niedrigen Erdorbits (LEO) von 400 bis 800 km fliegen. In diesen Höhen verdeckt die Erde in der Schattenphase bereits etwa ein Drittel des Sternenhimmels. Auf der Tagseite ist die Situation noch dramatischer. Die Sonne und die reflektierte Strahlung der Erde machen Aufnahmen des Sternenhimmels nur in einem sehr kleinen Bereich möglich. Je nach Position der Sonne macht der nicht überblendete Bereich nur etwa ein Sechstel des vollen Sternhimmels aus. Um die Detektionswahrscheinlichkeit zu erhöhen und auch aus Redundanzgründen, sollen mindestens zwei Sternsensoren verbaut werden. Da der Querschnitt eines CubeSats nur 10 cm x 10 cm beträgt, spielt die **Miniaturisierung** in der Entwicklung von Sternsensoren für Picosatelliten eine zentrale Rolle. Die Abmessungen eines Sternsensors inklusive Streulichtschutz (Baffle) sollen $30 \times 30 \times 80 \text{ mm}^3$ nicht überschreiten. Die Masse soll unter 50 g liegen.

Außerdem soll der OP1-Sternsensor eine **kostengünstige Lösung** bieten, da die meisten CubeSats low-cost Missionen sind. Dabei soll der Sensor die Genauigkeit von Präzisionssonnensensoren übertreffen und die Lage genauer als $0,01^\circ$ bestimmen können. Die Messungen sollen möglichst in allen Blickrichtungen entlang der drei Achsen stattfinden. Die Sternsensoren werden nur während präzisen Manövern bzw. Experimenten eingeschaltet. Die gewünschte Ausgaberate des Sensors soll maximal 2 Hz betragen. Dabei soll der Sternsensor nicht mehr als 250 mW verbrauchen. Die Lage soll bei Satellitendrehraten bis zu $1^\circ/\text{s}$ erfassbar sein. Diese Parameter sind für die Lageregelungssysteme der meisten CubeSats ausreichend.

Der Großteil der Picosatelliten wird im Durchschnitt für etwa ein Jahr Betrieb ausgelegt. Genau so ist die bevorzugte Lebensdauer des OP1-Sternsensors.

Zu obligatorischen funktionalen Eigenschaften gehören Software- und Sternkatalog-Upload sowie das Download des Rohbildes.

7.1.2 Optimierung 2 auf die Bedürfnisse von Nanosatelliten (OP2)

Aufgrund des größeren Volumens von Nanosatelliten ist die Miniaturisierung zweitrangig. Die OP2-Sternsensoren sollen hingegen ständig im operationellen Betrieb sein und die Lage mindestens vier Mal pro Sekunde bei gleicher Stromaufnahme wie OP1 ausgeben. Die Abmessungen des Sensorkopfs ohne Baffle sollen dabei $50 \times 60 \times 50 \text{ mm}^3$ nicht überschreiten. Die Konfiguration des Baffles ist missionsabhängig. So sollte z.B. für eine Referenzmission eines 24U CubeSats ein zylindrischer Baffle mit maximalen Abmessungen von $\text{Ø}80 \text{ mm} \times 100 \text{ mm}$ verwendet werden.

Die Bedürfnisse an Genauigkeit und zulässige Drehrate des Satelliten sind äquivalent zu denen von Picosatelliten.

Wie auch bei OP1 sind Speicherung und Download mehrerer Rohbilder in voller Auflösung sowie Softwareupdates vorausgesetzt.

Das Optimierungsziel bei OP2-Sternsensor ist, es einen möglichst leistungsfähigen und gleichzeitig energieeffizienten Sensor zu designen.

7.2 Anforderungen und Randbedingungen für OP1 und OP2

Die priorisierte Entwicklungsrichtung der beiden Sternsensoren OP1 und OP2 wurde in Form von Anforderungen quantitativ zusammengefasst (siehe Tabelle 18). Fett sind die primären Optimierungsziele markiert.

Tabelle 18. Anforderungen an die OP1- und OP2-Sternsensoren

Nr.	Anforderung	OP1	OP2
0010	Genauigkeit -pitch, yaw -roll	< 36 arcsec, < 120 arcsec 3 σ	< 36 arcsec, < 120arcsec 3 σ
0020	Brennweite	≤ 35 mm, ≥ 8 mm	≤ 35 mm, ≥ 8 mm
0030	Grenzmagnitude	$> 3,5^m, < 7,5^m$	$> 3,5^m, < 7,5^m$
0040	Sun excl. Angle	< 80°	< 40°
0050	Ausgaberate	≤ 2 Hz	\geq 4Hz
0055	Belichtungszeit	≥ 50 ms, ≤ 150 ms	≥ 50 ms, ≤ 150 ms
0060	Zulässige Drehrate	$\leq 1^\circ/s$	$\leq 1^\circ/s$
0070	Interface	CAN2.0B (I2C, RS422)	CAN2.0B (I2C, RS422)
0080	Stecker	Nano-D	Nano-D
0090	Output-Format	Quaternionen	Quaternionen
0100	Masse inkl. Baffle	< 50 g	< 120 g
0110	Abmessungen inkl. Baffle	< 30 x 30 x 80 mm	< 50 x 80 x 150 mm
0120	Spannungsversorgung	+ 4,5...15 V	+ 4,5...15V
0130	Durchschnittl. Verbrauch	250 mW	< 250 mW
0140	Op. Temperatur	-30...+50°C	-30...+50°C
0150	Lebensdauer	≥ 1 Jahr in LEO	≥ 1 Jahr in LEO
0160	Stückpreis	$\leq 10.000\text{€}$	$\leq 30.000\text{€}$

Die Anforderungen sind variabel und können im Laufe der Entwicklung immer weiter präzisiert werden. Die Randbedingungen dagegen entstehen außerhalb des Systems und kommen von außen. Sie sind nicht veränderbar und bleiben während der gesamten Entwicklung konstant. Die Randbedingungen haben meistens einen politischen, finanziellen, technologischen oder anwendungsspezifischen Hintergrund. Folgende Randbedingungen wurden für OP1 und OP2 in AROS-Vorhaben definiert:

Mechanik und Struktur

- Die Optik der beiden Sternsensoren muss ohne Demontage stufenlos manuell fokussierbar sein.

- Die Lage des OP1-Sternsensors relativ zur Struktur des Satelliten wird rein mechanisch, z.B. durch eine Passung, referenziert. Auf OP2 soll jedoch Platz bzw. eine Referenzfläche für einen optischen Alignment Cube vorgesehen werden.

Detektor und elektronische Komponenten

- Alle verwendeten elektronischen Komponenten müssen ITAR-frei sein.
- Für die verwendeten Bildsensoren muss ein technischer Support vorhanden sein.
- Die maximal zulässige Abnahmemenge der einzelnen elektronischen Komponenten beträgt 500 Stück.

Optik

- Maximale Anzahl der einzelnen Linsentypen ist auf 6 limitiert.
- Verkittete Linsen sind nicht erlaubt.
- Minimale Linsendicke beträgt 0,8 mm.
- Minimaler Linsendurchmesser beträgt 8mm.
- Auflagemaß liegt im Bereich von 3 mm bis 8 mm.
- Nur bevorzugte Gläsermaterialien nutzen (siehe Anhang E).

Elektronik

- Sternsensoren müssen mit unregelmäßiger Versorgungsspannung umgehen.
- Die Kommunikation soll über standardisierte Bus-Schnittstellen, wie I2C oder CAN2.0B laufen.

Einsatz- und Lagerungsumgebung

- Lagerungstemperatur liegt im Bereich von -40° bis $+85^{\circ}$ C.
- Vibration und Schock am Beispiel der Soyuz-2/Fregat Rakete.
- Strahlung: Total ionisation dose bis zu 100 Gy (10 kRad).

7.3 Elektrische Hardware

Um das automatische Designoptimierungstool vollständig parametrieren zu können, müssen einige Recherchen bezüglich der möglichen elektronischen Hardwareauswahl durchgeführt werden. Vor allem der Markt der Bildsensoren soll gründlich untersucht werden, um passende und vor allem beschaffbare Kandidaten zu finden. Für den Bildsensor wird auch eine geeignete Ausleseelektronik benötigt. Diese ist vom Detektorinterface und der Datenbitrate abhängig.

7.3.1 Bildsensoren

Der Bildsensor ist das Kernstück eines Sternsensors. Das gesamte System wird um den Bildsensor aufgebaut (siehe Kapitel 5.4). Je lichtempfindlicher der Bildsensor, desto größer ist die Grenzmagnitude der sichtbaren Sterne, die unter gleichen Bedingungen aufgenommen werden können. Genaue Angaben zur erreichbaren Grenzmagnitude sind für die Simulation des Sternsensors zwingend notwendig.

Wie bereits in Kapitel 4.5.2 beschrieben, haben moderne CMOS-Sensoren gegenüber der CCD-Technik Vorteile, die zur Miniaturisierung der elektronischen Hardware beitragen kön-

nen. Deshalb wurden die Marktrecherchen nur auf diesen Sensortyp begrenzt. Die Sensoren wurden anhand folgender Kriterien ausgewählt:

- Quanteneffizienz: > 60%
- Lichtempfindlichkeit: > 6,5 V/lux·s
- Auflösung: > 0,3 MPixel und < 2 MPixel
- Bildrate: > 10 fps
- Dynamik: > 50dB
- SNR: > 40%
- Bevorzugt werden monochrome Sensoren.

Nach einer detaillierten Untersuchung des Marktes kamen sieben CMOS Sensoren in die engere Auswahl (siehe Anhang F). Diese Sensoren wurden im Rahmen des AROS-Vorhabens genauer untersucht. Für jeden Kandidaten wurde eine Kamera bzw. ein Entwicklungsboard angeschafft, um deren Leistungsfähigkeit unter Schwachlichtbedingungen testen zu können. Dafür wurde ein künstlicher Stern in Kombination mit einem Industrieobjektiv bei einer festen Belichtungszeit von 100 ms verwendet. Die besten Ergebnisse lieferten die drei Sensoren EV76C454, AR0130 und ASX340CS. Die Eigenschaften aller drei Kandidaten sind in Tabelle 19 zusammengefasst. Weil ASX340CS ein Farbsensor ist, wird dieser nur als Reservekandidat behandelt.

Die beiden übrig gebliebenen Sensoren AR0130 und EV76C454 wurden für das Design der beiden Sternsensoren ausgewählt. AR0130 ist ein preisgünstiger Sensor mit Rolling-Shutter. Er soll in OP1 verwendet werden, weil es für Picosatelliten keine besonderen Anforderungen bezüglich Ausgaberate sowie der zulässigen Drehrate gab. Der EV76C454 hat einen globalen Shutter und kann bis zu 60 Bilder in eine Sekunde, bei einem sehr geringen Verbrauch von etwa 80 mW, aufnehmen. Wegen seiner Leistungsfähigkeit wurde dieser Sensor für den Einsatz in OP2 ausgewählt.

Tabelle 19. Spezifikation der besten Bildsensoren

Parameter	Einheit	EV76C454	AR0130	ASX340CS
Typ		CMOS	CMOS	CMOS
Hersteller		E2V	onsemi	onsemi
Auflösung	Pixel	838x640	1280x960	728x560
Verschluss		Global	Rolling	Rolling
Chrominanz		Monochrom	Monochrom	RGB
Farbtiefe	Bit	8	12	10
Pixelgröße	µm	5,8 x 5,8	3,75 x 3,75	5,6 x 5,6
Format	mm x mm	4,9 x 3,7	4,8 x 3,6	4,077 x 3,136
Empfindlichkeit	V/lux.s	7	6,5	16,5
Spektrale Empfindlichkeit, QE	%	72	78	60
Spannung am analogen Ausgang	V	0,6	1,25	1,25
Bildfrequenz	fps	60	54	60
Dynamik	dB	> 51	82	74,8
Signalrauschen, SNR	dB	max. 40	44	46
Dunkelrauschen	LSB/s	8		
Dunkelstrom	e/s			200
Sättigung	e-	10000		
Mikrolinsen		ja	ja	
Energieverbrauch	mW	80	270	234
Temperatur	°C	-30...+60	-30...+70	-30...+70
Bauform, Gehäuse		µLCC48	iLCC48	BGA63
Größe	mm x mm	10x10	10x10	7,5x7,5
ITAR		nein	nein	nein
Verfügbarkeit		ab 160 Stück	ab 1 Stück	ab 1 Stück
Langzeitverfügbarkeit		unbekannt	neu	neu
Preis pro Stück	EUR	80	35	10
Evaluierungsboard	EUR	2000	1200	1200
Quelle		[E2V20]	[ONS19-1]	[ONS19-2]

7.3.2 Systemarchitektur OP1

Ausgehend vom Bildsensor AR0130 und unter Berücksichtigung der Vorgaben zur Miniaturisierung und niedrigen Systemkosten wurde eine möglichst einfache Hardwarearchitektur, bestehend nur aus einem Mikroprozessor und einem Flash-Speicherbaustein mit seriellem SPI Interface, ausgewählt (siehe Abbildung 79).

Diese Architektur bringt eine Einschränkung mit sich: der Mikroprozessor muss eine integrierte parallele Bildsensor- oder Kameraschnittstelle haben. Auch die Taktrate des Prozessorkerns und seiner Peripherie muss der Auslesegeschwindigkeit des Bildsensors entsprechen. Als geeignet haben sich die Prozessoren der F4-Serie des Herstellers ST Semiconductor mit Cortex-M4 Kern erwiesen. Aus dieser Serie wurde der Prozessor STM32F407 für die OP1-Hardware selektiert. Er kann mit bis zu 168 MHz getaktet werden und besitzt alle notwendigen Schnittstellen. Angesteuert durch den Prozessor STM32F407 kann der OP1-

Sternsensor, bei voller Bildauflösung von 1,22Mpixel des AR130-Bildsensors, die angeforderte Ausgaberate von 2 Hz mit etwas Spielraum nach oben erreichen.

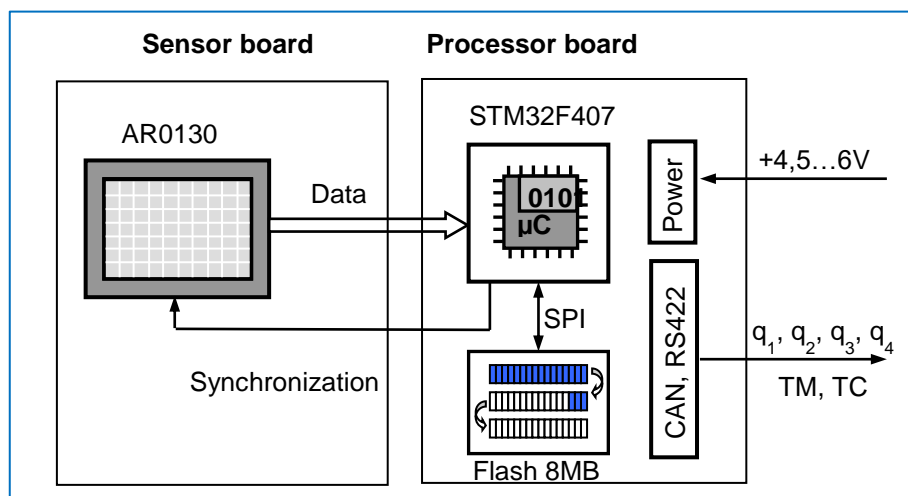


Abbildung 79. Hardwarearchitektur OP1

7.3.3 Systemarchitektur OP2

Der OP2 Sternsensor ist vor allem für die Anwendung in Nanosatelliten gedacht. Einer der Hauptforderung an OP2 ist die höhere Ausgaberate der Lage und ein zugleich niedrigerer Stromverbrauch. Um die höhere Ausgaberate erreichen zu können, muss die Bildverarbeitung inklusive Sternerkennung auf einen parallelarbeitenden und leistungsfähigen Logikbaustein, wie z.B. einen FPGA, ausgelagert werden. Die weiteren rechenintensiven Operationen für Sternidentifikation und Lageberechnung können dann problemlos auf einem Mikrokontroller in der verbliebenen Restzeit durchgeführt werden. Beste Voraussetzungen für eine solche Aufgabenverteilung liefert eine System-on-Chip-Lösung, wie z.B. SmartFusion-2 von Microsemi (MicroChip). Diese Chips haben sowohl einen FPGA als auch einen Cortex-M3 Mikroprozessor in einem Gehäuse integriert. Die Kommunikation zwischen beiden Logikeinheiten sowie mit den externen Speicher-Bausteinen erfolgt mittels AHB-Bus. Als nicht flüchtiges Speichermedium werden zwei MRAM Bausteine mit parallelem Interface verwendet. Dank parallelem Interface sind die Speicherzugriffe von OP2 um ein Vielfaches schneller als bei OP1.

Die Architektur der Hardware des OP2-Sternsensors ist im Vergleich zur OP1 wesentlich komplizierter. Durch die vorgestellte Hardwarekonfiguration (siehe Abbildung 80) kann der OP2-Sternsensor eine theoretische Ausgaberate von bis zu 10 Hz erreichen.

Der Programmierungsaufwand ist beim OP2-System doppelt so hoch, da hier effektiv zwei Bausteine FPGA und Mikroprozessor, programmiert werden müssen. Außerdem müssen beide Softwarekomponenten miteinander synchronisiert werden.

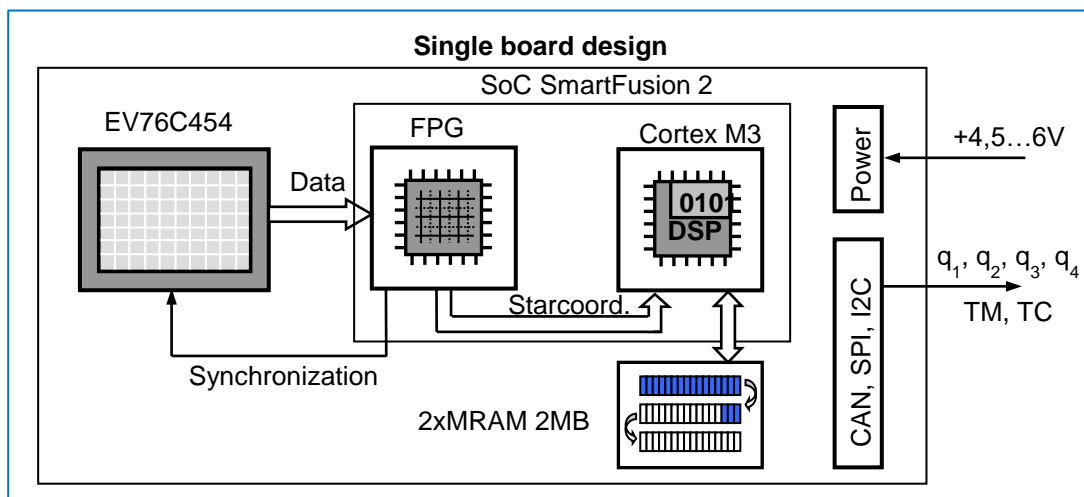


Abbildung 80. Hardwarearchitektur OP2

7.4 Ergebnisse der Optimierung

Da die geometrischen Eigenschaften der beiden ausgewählten Sensoren vergleichbar sind, ist eine getrennte Optimierung und draus folgende Herstellung der Optik zeitlich nicht effizient und wirtschaftlich nicht sinnvoll. Aus diesem Grund wurde die vollständige Optimierung nur einmal für beide Systeme durchgeführt. Dabei wurden die Anforderungen und Randbedingungen für OP1 und OP2 zusammengefasst.

Zahlreiche Optimierungsdurchläufe mit unterschiedlichen Einstellungen des evolutionären Optimierungsalgorithmus wurden unternommen. Populationsgröße, Anzahl der Nachkommen, Generationenanzahl, Selektions- und Rekombinationstyp wurden dabei variiert (siehe Tabelle 20). Bei allen Simulationen wurde die Anzahl der Linsen fest auf fünf gesetzt. Die Kombinationen mit vier Linsen haben eine viel zu schlechte Abbildungsqualität und Systeme mit sechs Linsen sind in der Regel viel zu groß.

Etwa 60 Durchläufe haben eine gültige Lösung gefunden. Populationen mit 50 bis 100 Individuen haben ein günstiges Laufzeit-Performance-Verhältnis und liefern verwertbare Ergebnisse. Die „Plus“-Selektion und elitäre Rekombination führte oft innerhalb von wenigen 10 Generationen zu schneller Konvergenz in einem lokalen Optimum. Die „Komma“-Selektion in Kombination mit zufälliger Partnersuche verteilt die Individuen zwar im Suchraum, sorgt aber für viele ungültige Lösungen, die die Optimierung verlangsamen. Tabelle 20 enthält einen Auszug der besten Ergebnisse aus der gesamten Optimierungssequenz. Die fünf Lösungen mit höchster Fitness, die in der Tabelle fett markiert sind, wurden zur Weiterentwicklung ausgewählt. Die Linsensysteme dieser Lösungen wurden im Rahmen des AROS-Vorhabens professionell bewertet und unter Berücksichtigung von Fertigungsaspekten validiert. Jedes System wurde erneut mit einem industriellen CAD-Tool für die Optikentwicklung simuliert. Qualitätskriterien wie Queraberrationen, Spotbilder, Übertragungsfunktionen oder Verzeichnung wurden genau verglichen. Es stellte sich heraus, dass nur Lösung 2 die benötigte Abbildungsqualität erreichen kann. Daher wurde diese für den Einsatz in den AROS-Sternsensoren ausgewählt.

Nr.	Parameter der Optimierung					Ergebnisse der Optimierung															
	Anzahl der Generationen	Anzahl der Individuen	Anzahl der Nachkommen	Selektion	Elternauswahl	Brennweite [mm]	Auflagemaß [mm]	Öffnung	Spiegelgröße [mm]	Spot-Asymmetrie	Distorsion [mm]	Vignettierung [%]	Min. Anzahl der Sterne	Ø Anzahl der Sterne	Datenbankgröße [byte]	Anzahl der Ringe	Anzahl der Speicherbegriffe	Grenzmagnitude	Belichtungszeit [ms]	Dauer der Simulation [s]	Fitness
1	10	100	x7	plus	zufällig	9,2744	3,5970	4,1466	0,0522	0,0291	-0,0027	100	44	67,7857	341,7480	690	1,4214e4	5,8312	847,7278	33887	717,1026
2	400	50	x7	plus	elitär	16,9467	4,5080	1,5186	0,0215	0,0502	0,0020	100	14	27,1905	452,3809	1034	4,5880e3	6,0934	84,7884	11671	768,4140
3	200	50	x7	plus	elitär	14,7859	7,0034	1,2066	0,0685	0,0414	0,0018	100	27	48,7619	1,0724e3	1034	2,2895e4	6,3899	92,3970	6374	799,2180
4	100	50	x15	koma	zufällig	12,2182	5,0834	2,0034	0,0212	0,0411	0,0020	100	9	20,7143	169,2344	1034	2,2206e3	5,2599	131,7420	54330	781,9263
5	10	50	x15	plus	elitär	9,5782	6,9438	4,4321	0,0430	0,0110	-0,0021	100	6	13,8571	29,8641	1034	137,0716	4,5880	130,1660	47362	687,4952
6	100	50	x7	koma	zufällig	10,9556	6,2668	4,2550	0,0309	0,0244	-0,0017	100	5	15,0952	46,5254	1034	273,3656	4,8279	124,1246	142005	683,3570
7	100	500	x15	koma	zufällig	9,4334	3,6643	2,2960	0,1271	0,0603	-0,0016	100	31	51,3571	212,9688	1034	4,3487e3	5,6033	99,5677	1868350	715,3541
8	100	50	x15	koma	zufällig	9,7214	5,5407	1,9259	0,0229	0,0244	-0,0019	100	14	26,7857	133,5684	1034	2,0190e3	5,1035	149,6715	204313	839,1950
9	100	50	x15	koma	zufällig	8,7696	5,5178	3,2071	0,0479	0,0117	-0,0018	100	17	33,8810	80,9902	1034	1,0845e3	5,1151	143,3849	255849	446,6044
10	100	50	x15	plus	elitär	10,8427	4,1614	4,3245	0,0560	0,0079	0,0018	100	6	17,1190	54,3613	1034	367,2563	4,9141	141,7229	237296	784,1808
11	200	25	x7	plus	elitär	13,0366	5,0843	4,0901	0,0684	0,0074	-0,0019	100	7	19,1429	109,3145	1034	955,0532	5,2654	121,1919	95480	682,8052
12	400	25	x7	plus	elitär	8,1157	4,1811	1,6272	0,3591	0,0589	-0,0133	100	89	145,9048	764,2480	115	4,0533e5	6,30073	129,2292	223962	653,1949
13	400	25	x7	koma	zufällig	9,2048	3,2309	2,2131	0,1138	0,0196	-0,0018	100	33	53,8095	212,9688	1034	4,4818e3	5,6022	97,0649	188162	745,1821
14	400	25	x7	koma	elitär	9,7007	3,1205	1,7415	0,1535	0,0403	-0,0069	100	72	116,4524	1030,8000	230	2,322e5	6,4000	113,0936	237635	653,7893
15	200	25	x7	plus	zufällig	10,8452	6,5435	4,6367	0,0443	0,0063	-0,0015	100	3	10,6190	26,7852	1034	88,9749	4,5128	112,5203	88197	629,2839
16	200	25	x7	plus	elitär	10,2875	3,1071	3,5867	0,0469	0,0204	0,0019	100	16	29,1190	109,3145	1034	109,3145	5,2625	149,2582	126347	573,7019
17	200	25	x7	plus	elitär	8,2404	4,2699	1,5746	0,2827	0,1073	0,0168	100	89	144,5714	762,6035	90	5,2729e5	6,3247	119,2661	51823	688,5541
18	200	25	x15	plus	elitär	11,9186	3,7369	2,3682	0,2086	0,0421	0,0031	100	70	110,9286	2102,9000	517	2,2286e5	6,7119	184,2131	268399	716,4166
19	200	25	x15	koma	elitär	12,8940	3,9728	2,1529	0,0210	0,0786	0,0017	100	15	31,7857	291,2510	1034	1,5763e5	5,7150	207,7349	311364	783,7275
20	200	25	x15	koma	elitär	11,5724	6,0445	2,2810	0,1597	0,0238	0,0051	100	78	127,3571	2324,7000	296	4,7662e5	6,7859	194,0415	211264	579,1911
21	100	25	x15	koma	zufällig	16,5420	7,1004	7,3435	0,0500	0,0262	0,0020	100	1	10,2857	82,7500	1034	370,5800	5,1280	213,8042	137752	656,6512

Tabelle 20. Ergebnisse der Optimierung für Systeme mit fünf Linsen

Fläche	1	2	3	4	5	6	7	8	9	10	11	12	13
t	∞				0,9	0,1	5,89	2,9	1,45	0,9	1,0	0,95	16,94
c [1/mm]	0	0,1031	2,5	-0,0303	0,0120	0	0,0769	-0,0435	-0,0971	-0,0602	-0,1377	0,0125	0
r [mm]		6		6	6	4,665	5	5	5	5	4	4	
Glass	air	air	N-LAK9	air	SF-6	air	air	N-LAF34	air	N-SF6	air	N-PSK3	air

Tabelle 21. Linsensystem der ausgewählten Lösung 2

Die Angaben zu den einzelnen Linsen von Lösung 2 sind in Tabelle 21 zusammengefasst. Die Brennweite des gesamten Linsensystems beträgt 16,95 mm und die Öffnung 1,52. Der Strahlengang durch die Linsen, die während des Optimierungsprozesses generiert wurden, ist in Abbildung 81 dargestellt.

Die Weiterentwicklung und Fertigstellung der Optik der ausgewählten Lösung wurde in enger Zusammenarbeit mit der Firma Christian Skrobol Optik- und Softwareentwicklung durchgeführt. Das automatisch generierte Linsensystem wurde mit Hilfe eines in der Firma vorhandenen kommerziellen Optikentwicklungstools erneut, allerdings lokal, optimiert. Materialien der ersten, zweiten und vierten Linse sowie einige Krümmungen der optischen Flächen wurden angepasst. Damit wurde eine weitere Verbesserung der Abbildungsqualität erreicht. Die Brennweite verkürzt sich dabei unwesentlich und liegt nun bei 16,44 mm. Die Öffnung der Optik ändert sich auch unwesentlich und erreicht den Wert von 1,61.

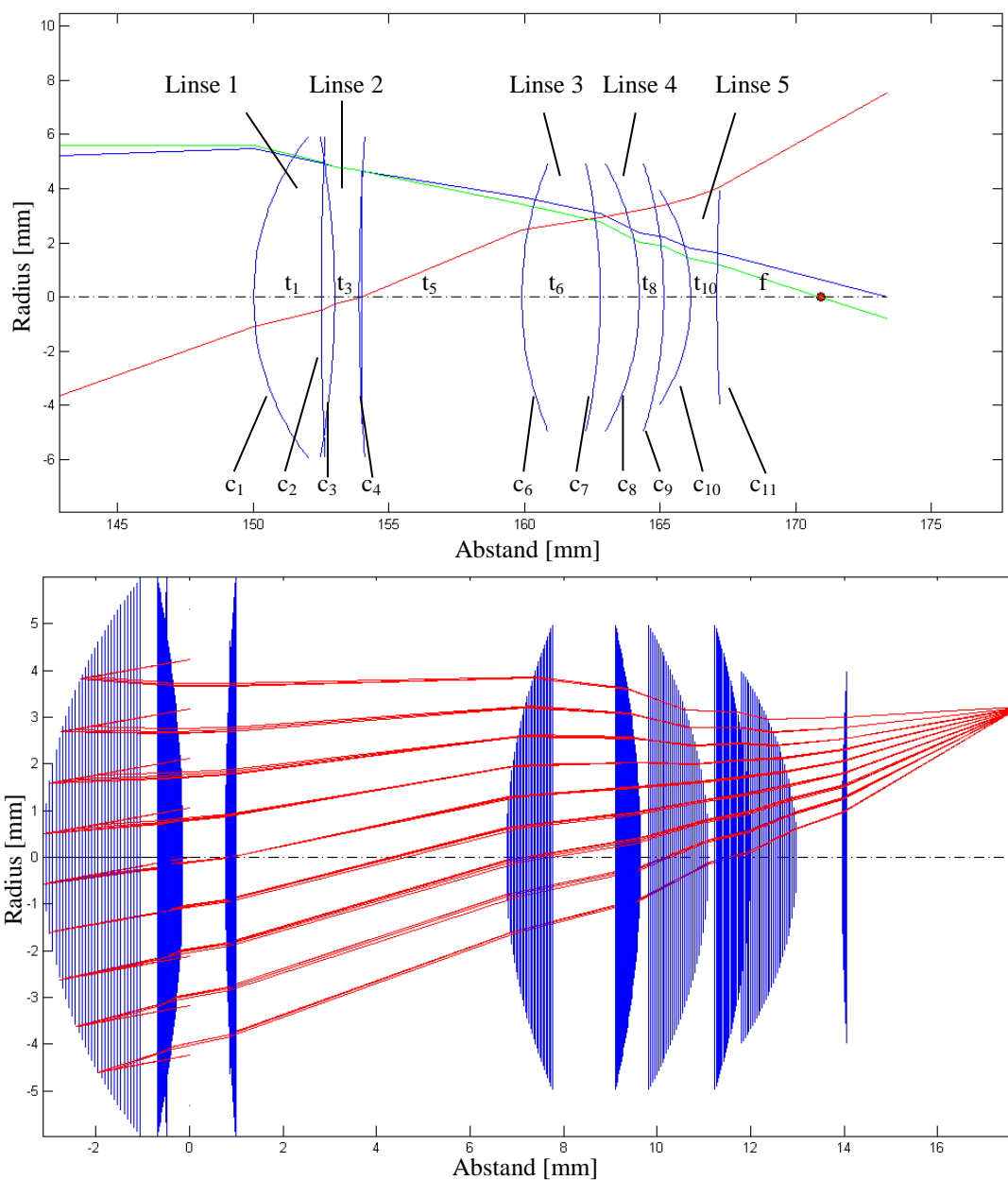


Abbildung 81. Paraxiale und Nichtparaxiale Strahlengang durch die ausgewählte Systemlösung

7.5 Bau und Test der Prototypen

Um die Funktionstüchtigkeit der vorgestellten Systemlösungen überprüfen zu können, wurde der automatisch erzeugte Entwurf der Sternsensoren finalisiert und fünf vollständige OP1-Systeme sowie drei Prototypen von OP2 gebaut und diversen Tests unterzogen.

7.5.1 Optik

Für das in Kapitel 7.4 ausgearbeitete Linsensystem wurde eine Toleranzrechnung durchgeführt. Diese Berechnung zeigte, dass das System empfindlich zur Einbaulage der einzelnen Linsen ist und eine sehr präzise Fertigung der mechanischen Komponenten mit einer Genauigkeit im $10\ \mu\text{m}$ - Bereich notwendig ist. Die mechanische Struktur wurde an der Professur für Raumfahrttechnik der Universität Würzburg nach Vorgaben der Optikentwickler der Fa. Skrobol entworfen. Das Objektiv besteht aus einem Tubus, zwei Zwischenringen und einem Vierkantgewindeflansch, der alle Linsen zusammenpresst. Das Objektiv wird anhand von vier Befestigungslöchern an den Sternsensor angeschraubt (siehe Abbildung 82). Die Fokussierung erfolgt anhand eines Gewinderings, der den Abstand zwischen dem Sensorgehäuse und dem Objektiv ändert.

Insgesamt wurden 10 Linsenpakete und 10 Sets der Mechanik gefertigt. Die Koordinierung der Produktion und den Zusammenbau der Objektive übernahm ebenfalls die Fa. Skrobol. Die fertig montierten Objektive wurden einzeln vermessen und Helligkeitsverlauf, Kontrast und Verzeichnung bestimmt. Alle Messergebnisse liegen im Bereich der Spezifikation.

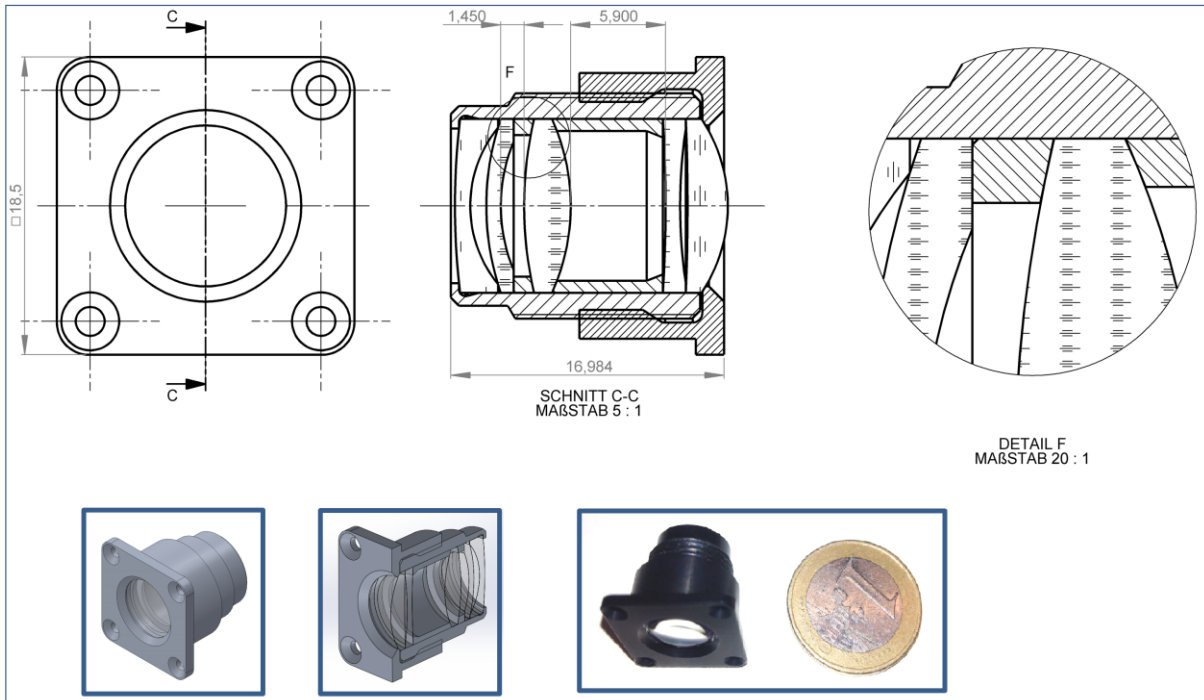


Abbildung 82. Objektiv der OP1 und OP2 Sternsensoren

Obwohl der erreichte Kontrast etwas niedriger im Vergleich zu den theoretischen Werten des CADs liegt, entsprechen alle gebauten Objektiven der Anforderungen bezüglich der Abbildungsqualität.

7.5.2 Mechanische Struktur

Die mechanischen Strukturen von OP1 und OP2 unterscheiden sich gravierend. Während OP2 eine klassische modulare Struktur, bestehend aus dem Sensor selbst und einem separatem, missions-spezifischen Baffle, hat, ist OP1 ein vollintegriertes System mit einem fest verbauten Baffle. Um OP1 möglichst kompakt gestalten zu können, wurden die elektronischen Komponenten auf zwei senkrecht zueinanderstehenden Leiterplatten verteilt. Daraus resultiert ein kompaktes quaderförmiges System mit Abmessungen von $21 \times 40 \times 55 \text{ mm}^3$ und einem Gewicht von 52 g (siehe Abbildung 83). Der Baffle besteht aus zwei Blenden mit einem rechteckigen Ausschnitt und bietet lediglich einen minimalen Streulichtschutz. Als mechanische Schnittstelle zur Befestigung des Sensors an die Satellitenstruktur sind vier M2,5 Gewindebohrungen auf der Rückseite der Befestigung des Sensorboards vorgesehen.

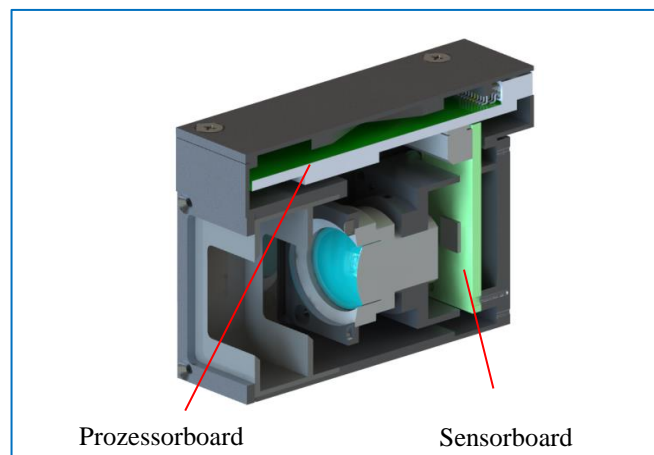


Abbildung 83. CAD-Modell des OP1-Sternsensors

OP2 ist ein modulares System, das im Gegenteil zu OP1 leicht an die Bedürfnisse der Kunden angepasst werden kann. Der Sensor selbst ist $51 \times 63 \times 30 \text{ mm}^3$ groß und 60 g schwer. Er besteht aus einer Elektronikleiterplatte, einem Gehäuse mit Deckel und dem bereits erläuterten Objektiv. Befestigt wird der OP2-Sensor an vier Bohrungen, angeflanscht am Gehäuse.

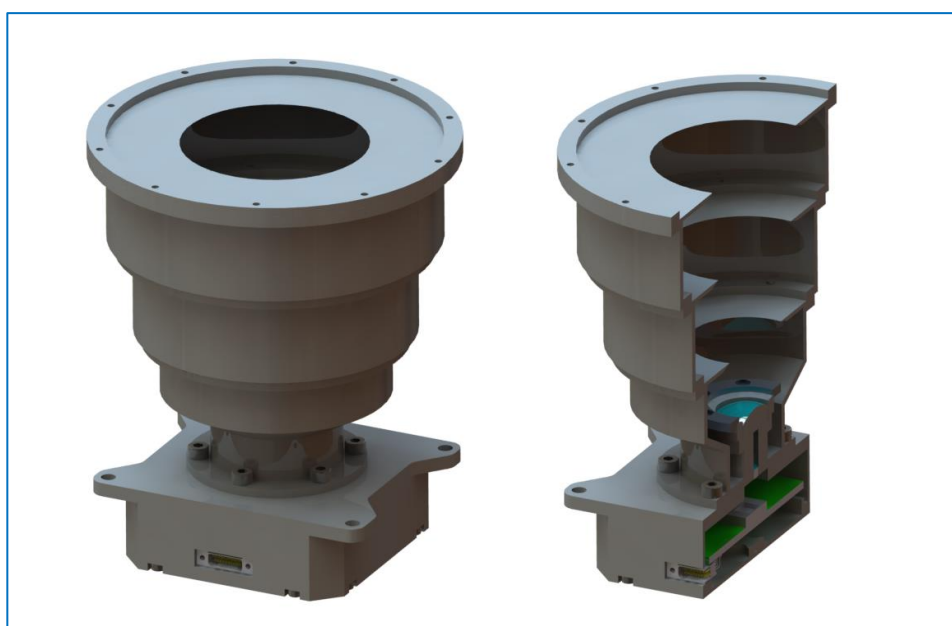


Abbildung 84. CAD-Modell des OP2-Sternsensors

Der in Abbildung 84 dargestellte Streulichtschutz ist lediglich ein Beispiel einer möglichen Blendenanordnung. Der Durchmesser dieses Baffles beträgt 70 mm. Er ermöglicht einen Sonnenausschlusswinkel von etwa 50°. Der Baffle kann aber für eine bestimmte Mission erneut konfiguriert werden.

Alle Komponenten von OP1 und OP2 wurden aus hochfestem Aluminium 7075 hergestellt und in Schwarz eloxiert.

7.5.3 Elektronik

Anhand den in Kapitel 7.3.2 und 7.3.3 vorgestellten Systemarchitekturen wurden die elektronischen Leiterplatten für OP1 und OP2 entworfen. Die elektronische Hardware von OP1 besteht aus zwei 6-lagigen Leiterplatten: Sensorboard und Prozessorboard. Diese sind anhand eines gewinkelten Steckers miteinander verbunden (siehe Abbildung 85). Das Sensorboard trägt den AR0130-Bildsensor und die dafür benötigte Peripherie. Das Prozessorboard akkommodiert Energieversorgung, Transmitterbausteine für die benötigten Schnittstellen, den STM32F407 Mikroprozessor und einen 128Mb SPI-Flash-Speicher.

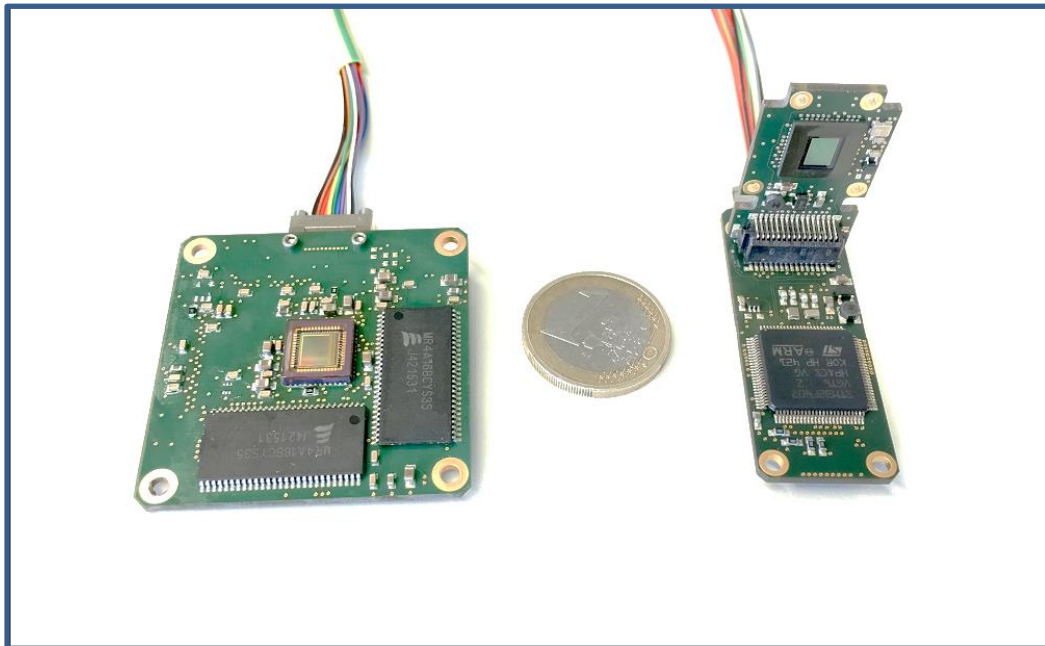


Abbildung 85. Elektronische Hardware von OP1 (rechts) und OP2 (links)

Die Leiterplatte des OP2-Sensors ist wesentlich komplexer und hat im Gegensatz zu den OP1-Platinen 8 Lagen. Auf der Vorderseite der Leiterplatte befinden sich zwei MRAM Speicherbausteine und der EV76C454 Bildsensor. Die Rückseite ist mit einem relativ großem Smart-Fusion2-Chip M2S005 im TQFP-Gehäuse mit 144-Pin bestückt. Wie bei OP1 sind auch hier Transceiver- und SPI-Flash-Bausteine untergebracht [GRE18].

Beide Systeme haben nach außen die gleiche Schnittstelle, eine 21-polige Nano-D Steckverbindung. Die Belegung der Pins ist ebenfalls identisch.

Alle Leiterplatten sind so konzipiert, dass keine besonderen Produktions-, Bestückungs- oder Löttechniken notwendig sind. Bei der Herstellung wurde das raumfahrttaugliche Laminat Panasonic R1755M verwendet. Die Oberfläche wurde im Chemisch-Zinn-Verfahren beschich-

tet. Es wurde ein ausgasungsarmer Lötstopplack in Grün verwendet. Die Leiterplatten wurden nach IPC-A-610 Klasse 2 manuell bestückt und in einem Durchlauf-Reflow-Ofen gelötet.

7.6 Funktionale Tests der Prototypen der Sternsensoren

Die fertig montierten OP1- und OP2-Sensoren wurden zuerst im Labor auf einem speziell dafür entworfenen Sternsimulator getestet. Der Simulator besitzt ein hochauflösendes Mikrodisplay und eine kollimierte Optik und kann einen beliebigen Ausschnitt des Sternenhimmels maßstabgetreu darstellen [SCH16-1]. Lediglich weicht die Helligkeit der Sterne und des Hintergrundes etwas von der Realität ab. Da bei der verwendeten Sternidentifikationsmethode keine Informationen über die Sternmagnitudo verwendet werden, ist der Simulator für funktionale Tests gut geeignet.

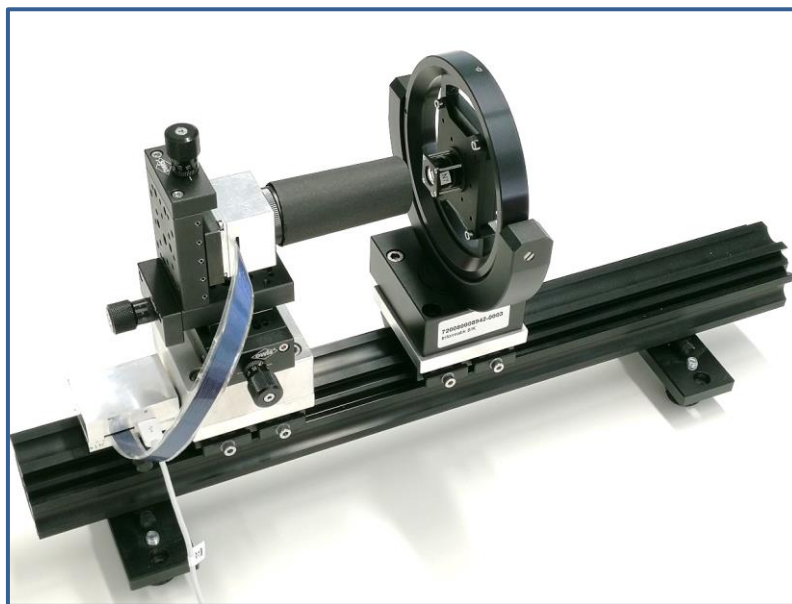


Abbildung 86. Hardware des Sternsimulators mit OP2

Am Simulator können Algorithmen zur Sternerkennung, Sternidentifikation und Lageberechnung unter verschiedenen Bedingungen getestet werden. So können Drehbewegung mit verschiedenen Raten und um drei Achsen simuliert werden. Auch die Grenzmagnitudo ist einstellbar und „falsche“ Sterne können sporadisch injiziert werden. Dadurch kann die Robustheit der Algorithmen überprüft werden.

Arbeiten zur Kalibrierung und zur Einstellung der Parameter der Sternidentifikation können aber nur am realen Sternhimmel durchgeführt werden, da die Kenntnis der tatsächlichen Grenzmagnitudo für die Datenbankgenerierung äußerst wichtig ist. Die Feldtests wurden in Dezember 2018 auf dem Gelände des Roque-de-los-Muchachos-Observatoriums in 2400 Meter Höhe durchgeführt. Dieser Ort ist für professionelle astronomische Beobachtungen optimal und zeichnet sich durch günstige Wetterverhältnisse sowie geringe Staub- und Lichtverschmutzung aus.

In den ersten Tests wurden die Grenzmagnitudo und der Schwellenwert in Abhängigkeit von der Belichtungszeit bestimmt. Beide Systeme wurden bei gleicher Referenzbelichtungszeit

von 100 ms verglichen. OP1 zeigte eine bessere Lichtempfindlichkeit und erreicht eine Grenzmagnitude von $6,5^m$. Die schwächsten Sterne wurden bei einem relativ niedrigen Schwellenwert von 17 (bei 8 Bit-Farbtiefe) erkannt. Die gute Empfindlichkeit hat OP1 dem Bildsensor AR130 zu verdanken. Dieser Bildsensor hat eine Spitzen-Quanteneffizienz von 78% und einen rauscharmen Rolling-Shutter. Dadurch ergibt sich ein helles und homogenes Bild. Bei OP2 betrag die Grenzmagnitude $6,2^m$ und der Schwellenwert bei der Erkennung lag deutlich höher bei 27.

Abbildung 87 zeigt zwei Beispielaufnahmen des Orion-Sternbildes. Links sind reale und farbinvertierte Aufnahmen des Orionnebels von OP1 bei einer Belichtungszeit von 1 s dargestellt. Das Systemkonzept von OP1 erlaubt nur die Speicherung eines kleinen Bildausschnitts von 230×230 Pixel. OP2 hingegen kann ein komplettes Bild aufzeichnen. In den realen und invertierten Abbildungen von OP2 ist im Hintergrund deutliches Rauschen zu sehen. Dies ist auf die Architektur des Global-Shutter-Sensors zurückzuführen.

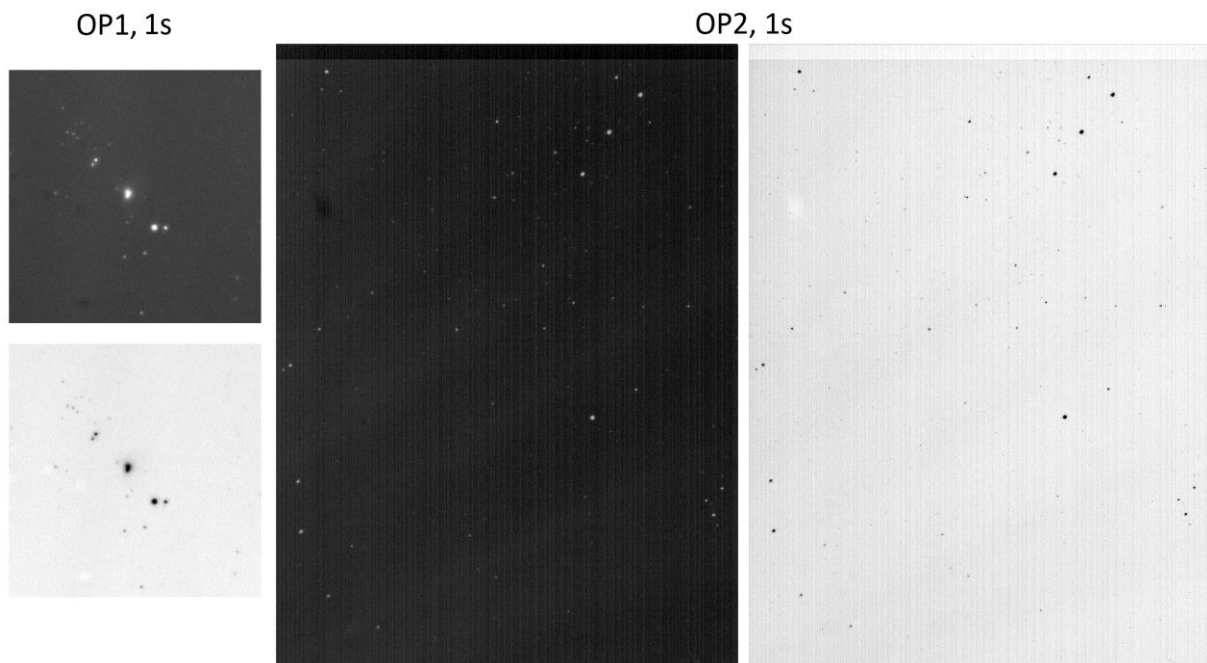


Abbildung 87. Beispielaufnahmen mit OP1 (links) und OP2 (rechts), jeweils Original und invertiert

7.7 Verbesserung der Sternidentifikation der AROS-Sternsensoren

Um die Häufigkeit von falschen Sternidentifikationen zu senken, müssen zwei wichtige Grenzparameter während der Feldtests eingestellt werden. Die Identifikation wird nur dann als valide betrachtet, wenn die Differenz zwischen Anzahl der erkannten Sterne und der theoretischen Mustergröße in vorgegebenen Grenzen liegt. Zusätzlich wurde eine weitere Maßnahme zur Verbesserung der Identifikation unternommen: die Sterne in der Nähe des Polsterns, die in den inneren Ringen des Kreismusters liegen, werden höher gewichtet. Die Gewichtungskoeffizienten wurden umgekehrt proportional zur Ringfläche gewählt, da die Wahrscheinlichkeit, dass die inneren Ringe besetzt sind, aufgrund ihrer kleineren Flächen niedriger ist.

Außerdem besitzen beide Sternsensoren ein Softwaremodul für eine automatische Kalibrierung der Verzeichnung, der Brennweite und des Offsets vom Mittelpunkt relativ zur optischen Achse. Diesem Softwaremodul liegt ein Simplex Algorithmus [COL04] und das Brown-Conrady-Korrekturmodell zugrunde. Nach dem Start der Autokalibrierung benötigen die Sensoren etwa 15 bis 45 Minuten, um die optimalen Einstellungen zu finden. Die Kalibrierung erhöht die Rate an richtigen Überstimmungen im Kreismuster um ca. 30% bis 50% im Vergleich zum nicht kalibrierten System.

7.8 Qualifikationsprogramm

Um die Weltraumtauglichkeit der OP1- und OP2-Sternsensoren prüfen zu können, wurden im Laufe des AROS-Vorhabens Umweltsimulationstest geplant und zusammen mit dem SONATE-Satelliten (siehe Kapitel 8.1) durchgeführt. Dazu gehören Strahlungstests, Thermal-Vakuum-Tests (TV-Tests) und mechanische Belastungstests. Die Testspezifikationen wurden angelehnt an den ECSS-E-ST-10-03C Standard erstellt.

Bei einem Strahlungstest wird die Strahlungsumgebung in Orbit simuliert. Abhängig von der Dicke der Aluminiumabschirmung liegt die erwartete Gesamtdosis ionisierender Strahlung (TID) für den Referenzorbit von 550 km etwa bei 2,5 kRad pro Jahr. Der Strahlungstest der Sternsensoren wurde im Dezember 2018 im Helmholtz-Zentrum Berlin durchgeführt. Als Quelle wurde ein Kobalt-60-Isotop verwendet. Die Probanden wurden 24,5 Stunden bestrahlt. Dabei wurde eine TID von 9,8 kRad ohne jeglichen Schaden an der Hardware erreicht. Die Standfestigkeit des Systems gegen die hochenergetische Teilchenstrahlung wurde vorerst nicht getestet.

Der TV-Test wurde in der TV-Kammer der Professur für Raumfahrttechnik an der Universität Würzburg ausgeführt. Dabei wurden vier Temperaturzyklen gemäß der Spezifikation in Tabelle 22 gefahren. Auch der TV-Test wurde erfolgreich abgeschlossen.

Tabelle 22. Spezifikation des TV-Tests auf Qualifikationsniveau

TV-Test	Temperatur	Vakuum	Anzahl der Zyklen	Verweilzeit
Nicht Operationell	-30°C bis +100°C	<10 ⁻⁵ mBar	4	2 Stunden
Operationell	-30°C bis +65°C			
Einschalttemperatur	-30°C bis +85°C			
Bedingungen	Temperaturgradient < 20°/Min.			

Die mechanischen Tests simulierten die Belastungen während eines Raketenstarts inklusive Separation der einzelnen Raketenstufen. Die mechanischen Tests für die Sternsensoren bestanden aus quasistatischen Belastungen, sinusförmigen Vibrationstests, zufälligen Vibrationsbelastungen und Schock-Tests. Das Belastungsniveau der mechanischen Qualifikation wurde anhand der Spezifikation des SONATE-Startanbieters festgelegt [EXO19] und sind an den Angaben im Raketenhandbuch für Soyuz 2.1-b angelehnt (siehe Tabelle 23 - Tabelle 27).

Tabelle 23. Spezifikation der quasistatischen Tests

Test 1 (2)	Z-Achse	X,Y-Achse
Frequenz [Hz]	0,25*Eigenfrequenz	
Beschleunigung [g]	7,7 (10)	3,8 (5)
Dauer [s]	180 (3)	30 (3)

Tabelle 24. Spezifikation der Sinus-Vibrationstests

Richtung	Frequenz [Hz]			
	1-2	2-5	5-10	10-20
	Sinusamplitude [g]			
Alle Achsen	0,3-0,5	0,5	0,5-1,0	1,0
Dauer [s]	600			

Tabelle 25. Spezifikation der kurzen Random-Vibrationstests

Test 1	Frequenz [Hz]						
	20-50	50-100	100-200	200-500	500-1000	1000-2000	G _{rms} (g)
	Power Spectral Density [$10^{-2}g^2/Hz$]						
Alle Achsen	2	2	2-5	5	5-2,5	2,5-1,3	7,42
Dauer [s]	120						

Tabelle 26. Spezifikation der langen Random-Vibrationstests

Test 2	Frequenz [Hz]						
	20-50	50-100	100-200	200-500	500-1000	1000-2000	G _{rms} (g)
	Power Spectral Density [$10^{-2}g^2/Hz$]						
Alle Achsen	2	2	2	2-0,8	0,8-0,4	0,4-0,2	3,58
Dauer [s]	480						

Tabelle 27. Spezifikation der Schock-Tests

Frequenz [Hz]	100-200	200-500	500-1000	1000-2000	2000-5000
Beschleunigung [g]	30-60	60-255	255-750	750	750-1500

Um die Integrität der Prüflinge überprüfen zu können, wird eine Resonanzsuche mit 0,5g von 5 Hz bis 2000 Hz vor und nach jedem Vibrationstest und in alle drei Achsenrichtungen durchgeführt. Abschließend erfolgt die Funktionsprüfung jedes Prüflings.

Die Vibrationstests wurden auf einer 35 kN Vibrationsprüfanlage und die Schocktests auf einem Pyroschock-Prüfstand der Firma Astro- und Feinwerktechnik Adlershof an zwei aufeinander folgenden Tagen Anfang Dezember 2018 ausgeführt. Die Sternsensoren haben alle mechanischen Tests erfolgreich bestanden.

7.9 Technische Spezifikation der OP1- und OP2-Sternsensoren

Der erfolgreiche Abschluss des Qualifikationsprogramms zeigt, dass keine weiteren technischen Änderungen bzw. Nachbesserungen der Sensoren notwendig sind. Somit stehen alle Systemcharakteristiken fest. Diese sind in Tabelle 28 zusammengefasst. Beide Systeme erfüllen die Anforderungen und Randbedingungen, die von den potenziellen Nutzern gestellt und in Kapitel 7.2 präsentiert wurden. Um einen direkten Vergleich mit anderen Sternsensoren für Pico- und Nanosatelliten ziehen zu können, wurde nach kommerziell verfügbaren Sternsensoren recherchiert (siehe Anhang H). Die herausragenden Eigenschaften gegenüber den Wettbewerbern sind fett markiert.

Tabelle 28. Technische Spezifikation der OP1- und OP2-Sternsensoren

Parameter	OP1	OP2
Geschätzte Genauigkeit: - pitch, yaw - roll	36 arcsec, 3 σ 180 arcsec, 3 σ	36 arcsec, 3 σ 180 arcsec, 3 σ
Brennweite	16 mm, F 1,6	16 mm, F 1,6
Grenzmagnitude	6,5	6,2
Sun excl. Angle	80°	45°
Ausgaberate	Max. 2 Hz	Max. 10 Hz.
Winkelgeschwindigkeit	Bis zu 1,5°/s	Bis zu 2,5°/s
Interface	CAN, RS422	CAN
Stecker	21-pol. Nano-D	21-pol. Nano-D
Masse	42 g	145 g
Abmessungen	21 x 40 x 54,5 mm³	51 x 63 x 92 mm ³
Spannungsversorgung	5V	5V
Verbrauch	300 mW	250 mW
Op. Temperatur	-20...+40°C	-20...+40°C
Flugerprob	SONATE 2019	

OP1 ist ein sehr kompakter und kostenoptimierter Sternsensor für Satelliten in der Größe eines einfachen CubeSat (1U) bis zu einem 12-fachen CubeSat (12U). Der Sensor wiegt 42g und benötigt 21x40x54 mm³ Bauvolumen. Somit passt er sogar in 4-facher Ausführung in den 10 cm - Querschnitt eines einfachen CubeSats. Das System basiert auf nur einem Mikrokontroller und hat einen relativ kleinen, auf die Anwendung angepassten, nicht flüchtigen Speicher. Die Ausgaberate des OP1-Sensors beträgt 2 Hz bei einer Leistungsaufnahme von etwa 300 mW.

Der OP2-Sensor ist leistungsoptimiert und kann die Lageinformation bis zu zehnmal schneller als OP1 bei sogar etwas geringerem Stromverbrauch liefern. Der Sensor ist etwas größer und für eine Satellitenklasse mit einer Größe über 12U optimiert. Das Herzstück des OP2-Systems ist ein System-on-Chip Prozessor, welcher die Vorteile von FPGA- und Mikroprozessor-Architektur fusioniert.

Dank innovativen Sternerkennungs- und Sternidentifikationsalgorithmen können beide Sensoren noch bei relativ hohen Drehraten die Lage mit etwas reduzierter Genauigkeitsperformance bestimmen.

8 In-Orbit-Verifikation auf SONATE

Der OP1-Sternsensor wurde an Bord des Nanosatelliten SONATE verifiziert. SONATE ist eine Technologiedemonstrationsmission, die an der Professur für Raumfahrttechnik der Universität Würzburg vorbereitet und durchgeführt wurde. Die Leitung des Vorhabens übernahm der Autor dieser Arbeit. Das Gesamtvorhaben wurde durch das Bundesministerium für Wirtschaft und Energie, vertreten durch die Deutsche Raumfahrtagentur DLR unter FKZ 50RM1606 gefördert. Das Hauptziel des Vorhabens ist die in-Orbit-Verifikation der Schlüsselemente zweier auch an der Universität Würzburg entwickelter hochautonomer Systeme.

Nach dreijähriger Entwicklung wurde der SONATE-Satellit am 05.07.2019 vom Kosmodrom Vostochny mit einer Soyuz Trägerrakete erfolgreich gestartet. Seit dem Start wird der Satellit fast täglich betrieben und diverse Experimente mit verschiedenen Nutzlasten wurden bereits durchgeführt. Die Dauer der Experiment-Phase beträgt ein Jahr. Nach Ablauf dieser Zeit geht der Satellit in Ausbildung der Studenten im realen Betrieb über.

8.1 SONATE-Satellit

Der SONATE-Satellit ist ein klassischer 3U-Cubesat mit einer Masse von 4,2 kg. Der gesamte Satellitenbus ist für die Erprobung und den Betrieb der autonomen Nutzlasten ASAP und ADIA im niedrigen Erdorbit ausgelegt [BAL17, WOJ17].

8.1.1 Nutzlasten

Eine von beiden Hauptnutzlasten ist ASAP-Light. ASAP-Light stellt eine auf Kernfunktionalitäten reduzierte Version des ASAP-Systems dar. ASAP ist ein autonomes Sensor- und Planungssystem, das unvorhersehbare Lichtphänomene, wie Meteore, Blitze, Sprites oder Blue Jets, im optischen Spektrum detektieren kann und weitere Aktivitäten wie Beobachtungsexperimente, Bodenkontakte oder Bildnachbearbeitung autonom, ohne Eingriff des Operators am Boden, planen kann [BAL13, WOJ14]. Ursprünglich wurde das ASAP-System im Rahmen eines DLR-Drittmittelvorhabens (FKZ 50RM1208) für einen 27U-Satelliten konzipiert, wiegt in voller Ausstattung etwa 3 kg und ist $175 \times 245 \times 100 \text{ mm}^3$ groß (siehe Abbildung 88). Es besitzt zwei unabhängige Kameras: eine, mit einem breiten Sichtfeld von $42^\circ \times 42^\circ$ (WFOV), und eine weitere, mit einem schmalen Sichtfeld von $4^\circ \times 4^\circ$ (NFOV). Die Bildverarbeitungselektronik ist für beide Kameras identisch und basiert auf einem Xilinx Spartan-6 FPGA. Aufgrund des beschränkten Bauvolumens eines 3U-CubeSats konnte nur das kleinere Kamerasystem mit WFOV an Bord von SONATE akkommodiert werden. Das reduzierte System, (ASAP-Light) ermöglicht Videoaufnahmen mit Auflösungen von bis zu 4 MPixel bei Bildraten bis zu 60 fps [BAL14-2]. Jedes Videobild wird durch vordefinierte Bildverarbeitungsalgorithmen analysiert, um interessante Ereignisse oder Lichtphänomene zu extrahieren. Die extrahierten Videosequenzen können in einem 64 GByte großen Flashspeicher dauerhaft abgelegt werden [WOJ13-1, WOJ13-2].

Eine weitere Besonderheit von ASAP-Light ist eine Planungssoftware, die den Missionsfortschritt in Abhängigkeit von den verfügbaren Ressourcen selbstständig verändern kann. Die Planungssoftware kann die nächsten Aktionen planen und neue Kommandolisten für den

OBC des Satelliten selbstständig erstellen. Das Planungssystem des ASAP-Light wird extern auf einem 1 GHz Sitara Mikroprozessor von Texas Instruments ausgeführt.

Die Leistungsaufnahme von ASAP-Light ist relativ groß und macht im Durchschnitt 8 W aus. Die Peak-Leistung kann bis zu 12 W betragen.

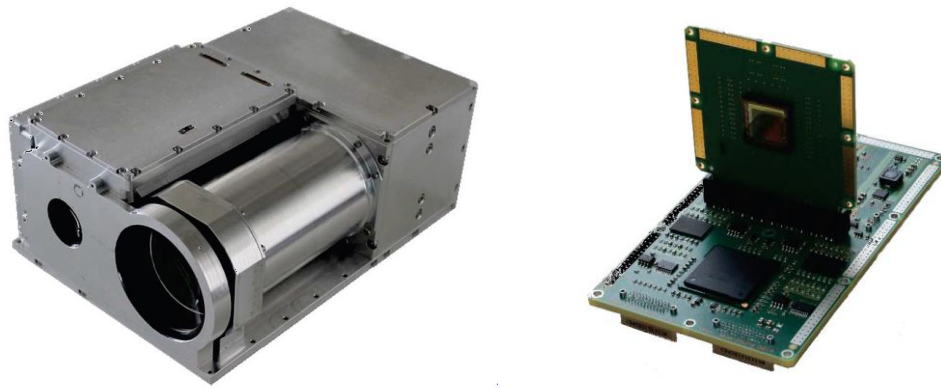


Abbildung 88. ASAP-System (Links) und ASAP-Light (rechts)

Die zweite Hauptnutzlast ist ADIA-Light. ADIA-Light ist ein modellbasiertes Diagnosesystem zur Bestimmung von Symptomen und Ursachen möglicher Fehlfunktionen an Bord des Satelliten und zur Vorhersage sich anbahnender Ausfälle [FEL13, FEL19]. ADIA-Light verwendet modellbasierte Verfahren in Kombination mit algorithmischer Graphentheorie, um Anomalien im Satelliten zu finden. Es empfängt alle Telemetrie-Frames vom OBC, um den aktuellen Zustand des Satelliten zu analysieren und Trends zu erkennen. Diese Telemetrie-Frames werden mit einem simulierten Satellitenmodell verglichen, um Abweichungen zu erkennen [DJE17]. ADIA-Light ist eine eingebettete Version der vollständigen ADIA++ Desktopsoftware (FKZ 50RM1524), die an SONATE angepasst wurde und nur die Kernfunktionen zur Symptomentektion enthält. ADIA-Light läuft auf der gleichen Hardware wie das ASAP-Light Planungssystem.

Beide Hauptnutzlasten haben erheblichen Einfluss auf Hardware- und Softwaredesign sowie auf den Betrieb des SONATE-Satelliten. Vor allem soll die Sicherheit des Satelliten während der autonomen Experimente gewährleistet werden [RAP18].

Außer den beiden Hauptnutzlasten sind mehrere sekundäre Experimente zur Technologiedemonstration an Bord. Zwei davon gehören zum ADCS: drei Reaktionsräder und zwei OP1-Sternsensoren sind in SONATE integriert und sollen im Orbit verifiziert werden. Zusätzlich sind zwei Slow Scan Television (SSTV) Module für Experimente im Kommunikationsbereich vorgesehen. Mit den SSTV-Modulen ist es möglich, in der Auflösung reduzierte Bilder von ASAP-Light oder von den Sternsensoren über die VHF-Funkstecke direkt an Funkamateure am Boden zu senden [SCH18-2].

8.1.2 Satellitenbus

Der Satellitenbus von SONATE ist ein robustes, voll redundantes und ein-Fehler-tolerantes System. Es wurde entsprechend den Hardwareanforderungen der autonomen Nutzlasten ASAP-Light und ADIA-Light entwickelt [BAL19, SCH16-2].

Eine Systemübersicht des Satelliten ist in Abbildung 89 dargestellt. Alle Komponenten sind über jeweils zwei redundante, 50-polige Konnektoren an eine Backplane angebunden. Über die Backplane werden alle Schnittstellen, logischen Signale und die Stromversorgung verteilt. Die Kommunikation zwischen den Komponenten erfolgt über ein Bussystem auf Basis von CAN2.0B. Zwei redundante, geregelte +5V Powerbusse versorgen den gesamten Satellit.

Die Backplane hat 11 Steckplätze für eine Interface-Leiterplatte, zwei PCUs, zwei Kommunikation-Leiterplatten, vier OBDHs, eine ADIA/ASAP-Leiterplatte und eine ADCS-Leiterplatte. Am Ende der Backplane ist eine zwölfte Leiterplatte angebracht. Diese verbindet alle Solarpaneele mit der Backplane und wird als Abschluss-Leiterplatte bezeichnet.

Eine zentrale Rolle bei der Steuerung aller Nutzlasten, ausgenommen von ADIA-Light, spielt die Interface-Leiterplatte. Sie bildet eine Brücke zwischen dem Satellitenbus und den einzelnen spezifischen Interfaces von ASAP-Light, der drei Reaktionsräder und der beiden OP1-Sternsensoren.

Im weiteren Verlauf des Kapitels wird kurz auf die Besonderheiten der einzelnen Subsysteme des Satelliten eingegangen.

8.1.2.1 On Board Data Handling (OBDH)

Eine spezielle OBDH Hard- und Software wurde entwickelt, die eine sichere Handhabung mit den autonomen Experimenten gewährleisten kann, ohne die Mission zu gefährden. Jede der vier identischen OBDH-Leiterplatten erhält eine von vier möglichen Rollen: On Board Computer (OBC) oder Payload Data Handling (PDH), als Master oder Slave. Die Rollenzuweisung wird selbstständig und dynamisch zwischen den Leiterplatten ausgehandelt.

Parallel zu den typischen Aufgaben wie Zeitmanagement, Orbitberechnung, Telemetrie-Sammlung und Ressourcenüberwachung verwaltet der OBC zwei Telekommandolisten: eine passive und eine aktive. In die passive Liste können entweder Time-Tag-Kommandos vom Operator am Boden oder die Kommandos von ASAP-Light, die während eines Experiments mit dem autonomen Planungssystem selbstständig erstellt werden, geschrieben werden. Diese kann nach erfolgreicher Validierung aktiviert werden. Das PDH managt die gesamte Nutzlast-Telemetrie von SONATE, einschließlich der Bilder und Videosequenzen von ASAP-Light, und bereitet die gesammelten Daten für eine Übertragung zur Bodenstation vor.

Die OBDH-Hardware basiert auf einem Cortex-M4-Mikroprozessor von ST Microelectronics, dem gleichen Typ wie OP1 (STM32F407). Für die Datenverwaltung stehen je OBDH 8 MByte flüchtiger SRAM- und 256 MByte nichtflüchtiger Flash-Speicher zur Verfügung.

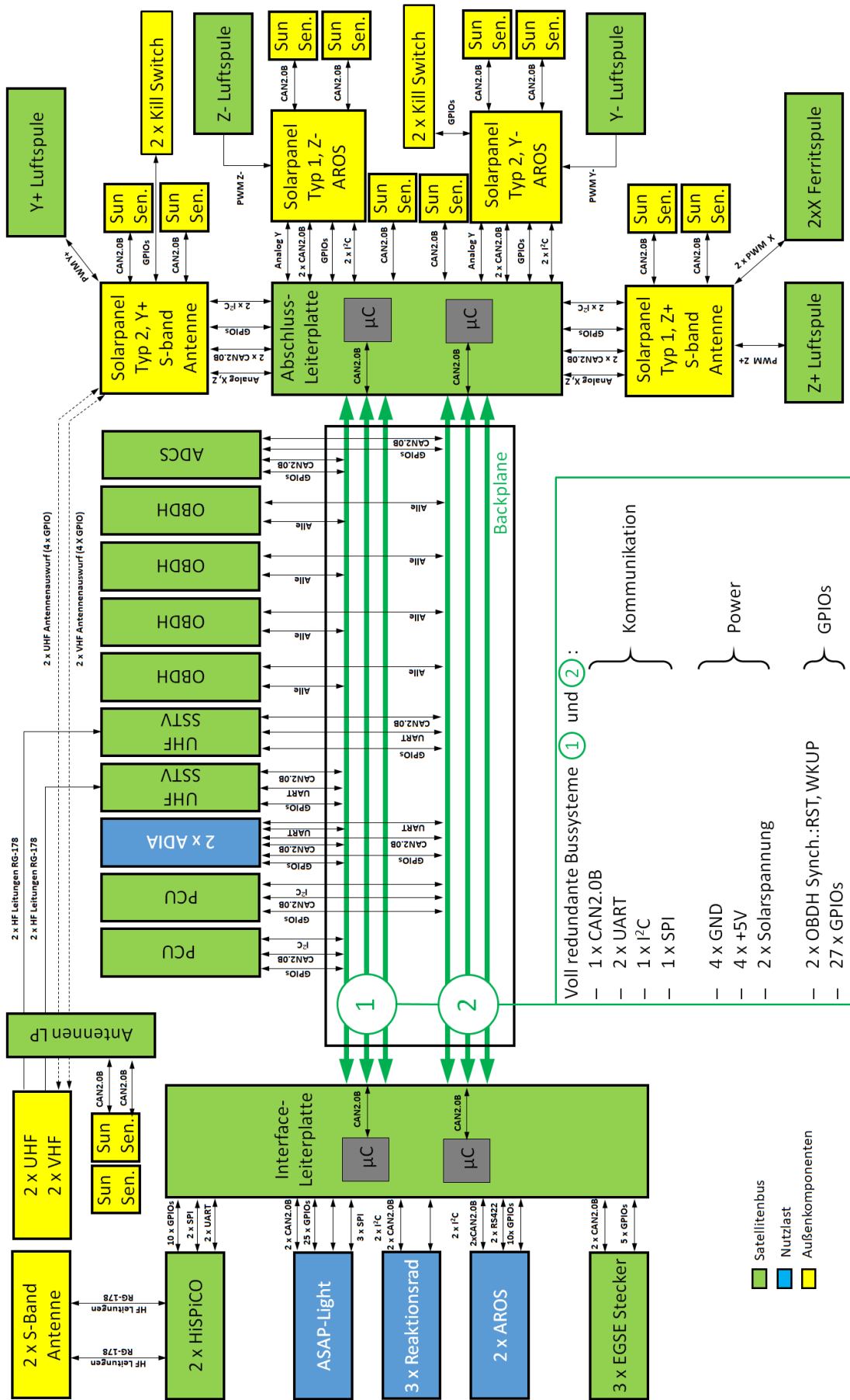


Abbildung 89. SONATE-Systemübersicht

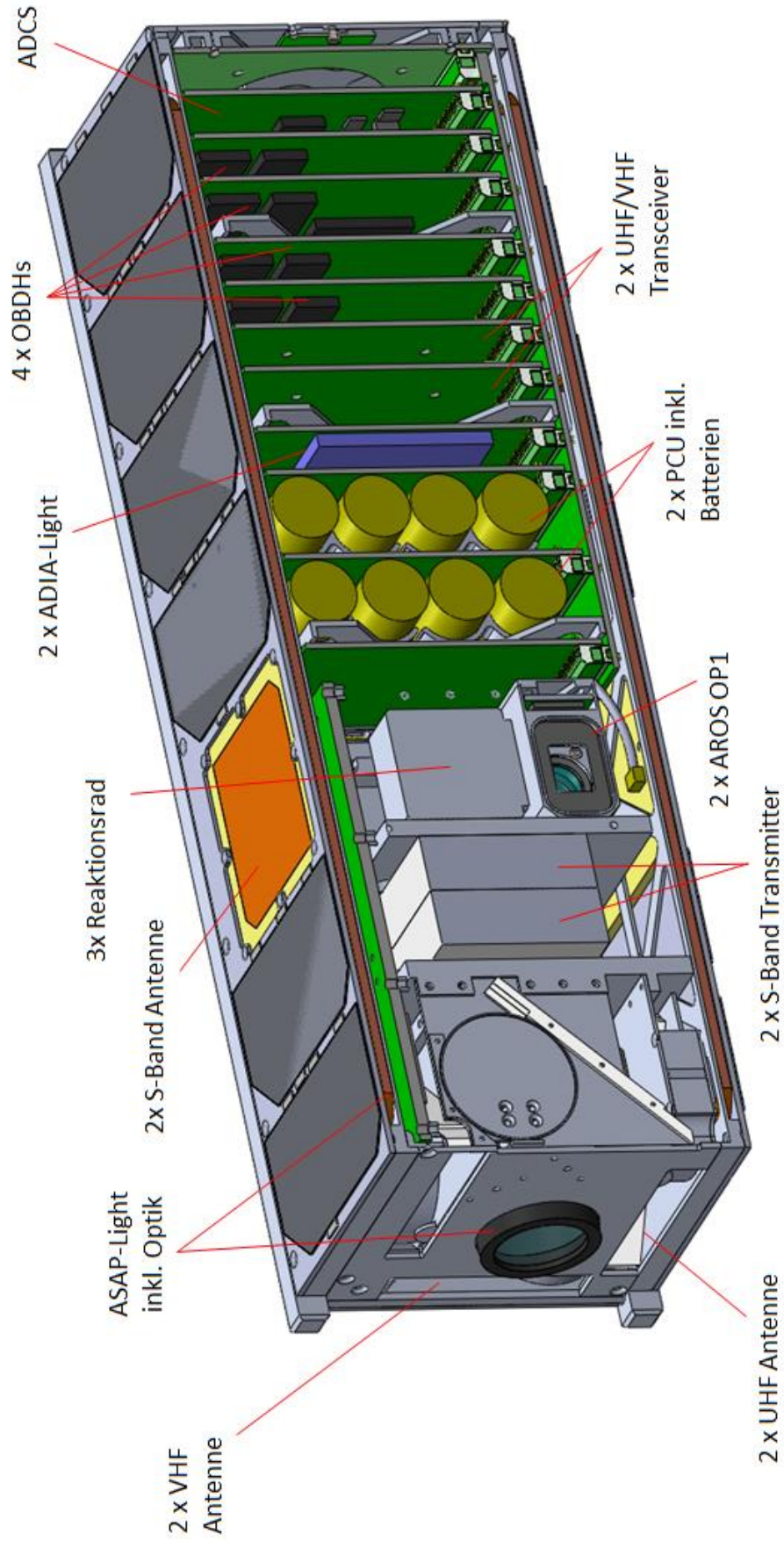


Abbildung 90. Akkommodation des SONATE-Satelliten

8.1.2.2 Kommunikationssystem

Der nominale Satellitenbetrieb erfolgt über eine UHF-Kommunikationsstrecke bei 437,025 MHz im Halbduplex-Modus mit einer Bitrate von 9,6 kbit/s. Dazu sind zwei kommerzielle UHF-Transceiver in redundanter Ausführung und zwei entsprechende Monopol-Antennen in den Satellitenbus integriert. Als Backup kann die Kommandierung auch über die VHF-Transceiver der SSTV-Module bei 145,840 MHz erfolgen. Für die Übertragung der Nutzlastdaten wird das S-Band bei 2268,700 MHz verwendet. Zwei HiSPiCO-Transmitter von der IQ wireless GmbH und zwei Patch-Antennen auf gegenüberliegenden Seiten des Satelliten sorgen für ein Downlink mit bis zu 1,6 Mbit/s. Passend zum Raumsegment wurden zwei Bodenstationen für UHF/VHF- und S-Band eingerichtet (siehe Kapitel 8.2).

8.1.2.3 Energieversorgung

Der SONATE-Satellit verfügt über zwei vollständig redundante +5V Versorgungsbusse. Die elektrische Energie wird für jeden Bus durch zwölf körperfeste GaAs-Solarzellen generiert. Jede Zelle ist mit einem separaten MPPT-Regler versehen. Je Bus werden vier parallel geschaltete Li-Ionen-Zellen mit einer Gesamtkapazität von 4 Wh zur Speicherung der elektrischen Energie verwendet. Die Spannung dieses Batteriepacks ist nicht konstant und ändert sich in Abhängigkeit vom Ladezustand. Deshalb wird ein Aufwärtsschaltregler für die Generierung der +5V Busspannung eingesetzt. Der Regler kann eine Spitzenleistung von bis zu 20 W und eine Dauerleistung von 8 W liefern. Diese Leistung wird z.B. für den Parallelbetrieb des S-Band-Senders und des ASAP-Light-Systems im Video-Capture-Modus benötigt.

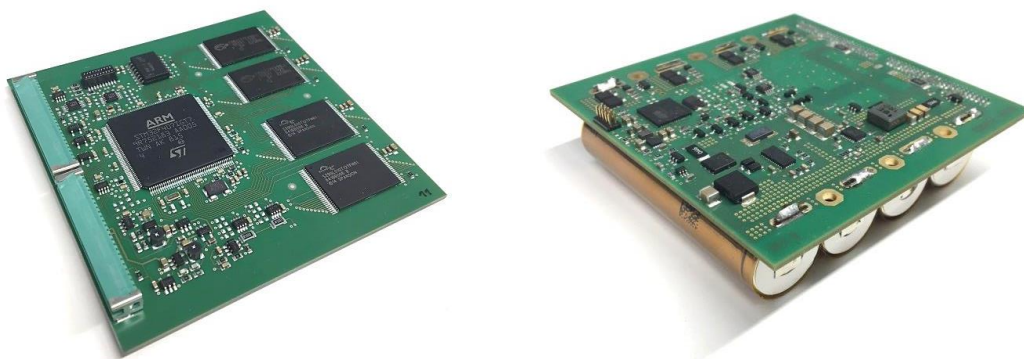


Abbildung 91. OBDAH- (links) und PCU-Leiterplatte (rechts)

8.1.2.4 ADCS

Das ADCS-Subsystem hat eine klassische Struktur wie bei den meisten CubeSats. Zur Lagebestimmung werden sechs digitale Miniatur-Sonnensensoren, einer pro Achsenrichtung, und dreiachsige Magnetometer eingesetzt. Die Drehraten des Satelliten werden mit dreiachsigen MEMS-Gyroskopen gemessen. Als Aktuatoren werden zuerst nur Magnetorquer verwendet: vier Luftspulen sind auf den Leiterplatten der Solararrays angebracht und zwei Ferritkernspulen sind im Inneren des Satellitenkörpers platziert, um die Achse mit den kleineren Flächen zu bedienen. Das ADCS hat einen eigenen STM32L4 Steuerungsmikroprozessor mit entsprechender Peripherieelektronik. Alle oben genannten ADCS-Komponenten wurden redundant in das Satellitensystem integriert.

An Bord von SONATE befinden sich drei Reaktionsräder und zwei OP1-Sternsensoren. Diese sind zunächst als Experimente zur Technologiedemonstration an Bord. Jedoch können diese

nach erfolgreicher Validierung in den Regelkreis eingekoppelt werden, um eine vollständige Drei-Achsen-Stabilisierung zu erreichen.

8.1.2.5 Struktur und Mechanismen

SONATE ist ein 3U-CubeSat mit standardisierten Abmessungen der Hauptstruktur von 100x100x340 mm³. Die mechanische Struktur nimmt nicht nur alle Nutzlast- und Buskomponenten auf, sondern sorgt auch für die Integrität des Satelliten während der gesamten Mission. Außerdem gleitet der Satellit während des Auswurfs aus dem Launch-Container auf vier äußeren Schienen.

Der SONATE-Satellit kann imaginär etwa in der Mitte an zwei Hälften geteilt werden (siehe Abbildung 90). Im Vorderteil sitzen die meisten Experimente: ASAP-Light, zwei OP1-Sternsensoren und drei Reaktionsräder. Die Optik von ASAP-Light ist von außen gut erkennbar (siehe Abbildung 92). Die beiden Sternsensoren schauen in entgegengesetzte Richtungen und haben einen freien Blick durch rechteckige Öffnungen in den Seitenpanelen zwischen der zweiten und der dritten Solarzelle. Hier befinden sich auch drei Stecker zum Anschluss des EGSE (Electrical Ground Support Equipment).

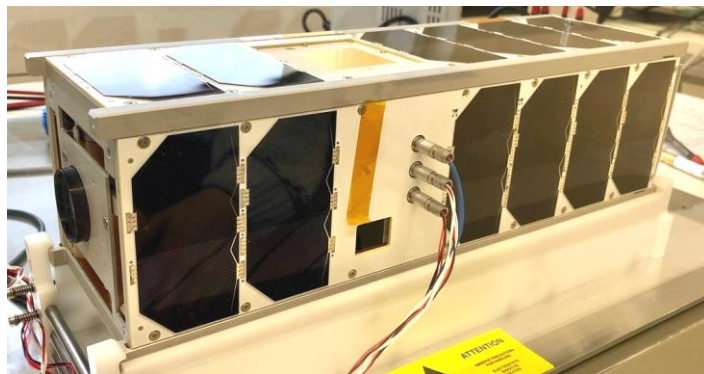


Abbildung 92. SONATE Qualifikationsmodell.

Im vorderen Bereich des Satelliten befinden sich vier aufgerollte Kommunikationsantennen. Die Länge der VHF-Antennen beträgt 50 cm. Die UHF-Antennen sind wesentlich kürzer mit nur 17,5 cm. Trotzdem benutzen sie den gleichen Auswurfsmechanismus. Nach dem Ausrollvorgang stehen die gleichlangen Antennen orthogonal zueinander.

8.2 Bodensegment und Satellitenbetrieb

Für die Kommandierung und den Empfang der Telemetrie wurden zwei Bodenstation für UHF/VHF- und S-Band auf dem Campus Hubland Nord der Universität Würzburg aufgebaut (siehe Abbildung 93). Analog zum Raumsegment werden zwei Amateurfunkfrequenzen für die Kommunikation mit dem Satelliten verwendet. Die verwendeten UHF-, VHF- und S-Band-Frequenzen wurden entsprechend zur BNA- und ITU-Vorgaben national und international koordiniert.



Abbildung 93. SONATE Missionskontrollraum und UHF/VHF-Bodenstation [SCH18-2]

Der SONATE-Satellit wird aus dem eigenen Missionskontrollzentrum der Professur für Raumfahrttechnik betrieben. Für den Betrieb stehen verschiedene Software-Tools zur Verfügung, die einen Operator bei seiner Tätigkeit vielseitig unterstützen. So können sowohl lange und komplizierte Experimente als auch Routine-Prozeduren bequem in ein Missionsplanungstool eingetragen werden. Aus der chronologischen Reihenfolge der eingegebenen Aktivitäten generiert das Tool anhand der TC-Datenbank eine gültige Kommandoliste. Der Operator kann dann die fertigen Telekommandolisten in einen Telekommando-Client laden und von hier als übersetzte Bytesequenz an einen der Bodenstationsserver senden. Der Telekommando-Client hat umfangreiche Bearbeitungs- und Kommandierungsfunktionen. So können neue Kommandos erstellt, kopiert oder überarbeitet werden. Unnötige Kommandos können gelöscht oder verschoben werden. Bei jeder Änderung oder bei einem Neueintrag wird die Korrektheit des Kommandoinhaltes, der Zeitstempel und die Konsistenz der ganzen Liste geprüft und mit der Datenbank verglichen. Der Operator hat eine Möglichkeit, Kommandos einzeln, in Gruppen, asynchron oder synchron mit Kommandozeiler zu schicken. Je nach dem für die Kommandierung gewählten Kommunikationsband wird der UHF- oder VHF-Server in der Bodenstation angesprochen. Der Server verpackt die entsprechenden Bytesequenzen in Pakete entsprechend dem AX.25 Standard. Die Pakete werden dann über ein Modem zum Funk-Transmitter weitergeleitet und über eine der beiden Antennen an den Satelliten übermittelt.

Der Empfang der Telemetrie erfolgt ähnlich nur in umgekehrter Reihenfolge. Das empfangene Funksignal kommt vom Receiver zum Modem und wird dort dekodiert. Die dekodierten Pakete werden dann vom UHF-Server entpackt und archiviert. Ein speziell entwickelter Telemetrie-Client verbindet sich mit dem Server und der TM-Datenbank, interpretiert und visualisiert die empfangenen Daten. Dadurch kann der Operator alle für den Betrieb notwendigen physikalischen Größen und Komponentenzustände direkt vom Bildschirm ablesen.

Weitere Hilfswerkzeuge zur Passvorhersage, Steuerung und Nachführung des Antennenrotors, Kompensation von Dopplerverschiebung, Funkgerätsteuerung, Messung der Sendeleistung erleichtern und automatisieren größtenteils den Betrieb.

8.3 Experimente mit den OP1 Sternsensoren

Bereits in der Inbetriebnahme-Phase im Zeitraum von September bis Oktober 2019 wurden die beiden OP1-Sternsensoren eingeschaltet und wichtige Telemetrie-Informationen über deren Zustand empfangen. Die Stromaufnahme der Sternsensoren lag im nominalen Bereich. Im weiteren Verlauf der SONATE-Mission wurden mehrere Versuche unternommen, ein Sternbild aufzunehmen und entweder analog über SSTV oder digital über die UHF-Strecke zu übertragen. Der Zweck dieser Aufnahmen war es, die Grenzmagnitude und den Detektionsschwellenwert im All zu bestimmen. Diese Werte sind abweichend von den am Boden ermittelten Werten, da die Atmosphäre diese beeinflusst. Eine digitale Beispielaufnahme, die am 06.02.2020 empfangen wurde, ist in Abbildung 94 zu sehen. Die schwarzen Streifen im Bild sind durch Verlust von Paketen bei der Übertragung entstanden. Wie deutlich zu sehen ist, hat der aufgezeichnete Stern eine längliche Form. Diese Verschmierung wurde durch eine Drehung des Satelliten verursacht. Anhand der Relation der Sternausdehnung zur Belichtungszeit von 100 ms wurde eine Drehrate des Satelliten von etwa $0,6^\circ/s$ errechnet. Der errechnete Wert stimmte mit der vom ADCS gemessenen Drehrate überein.

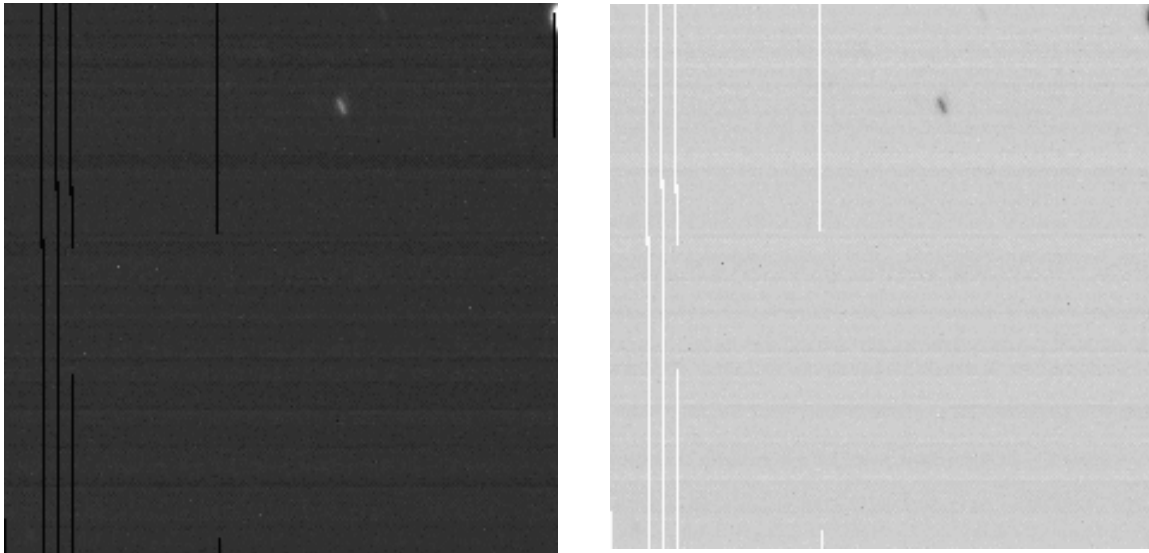


Abbildung 94. Originalaufnahme eines Sterns (links) und invertiertes Bild (rechts)

Durch die Analyse des Rauschpegels und der Sternhelligkeit im Bild konnte der Schwellenwert ermittelt werden. Ein Schwellenwert von 17 bei 8 Bit Farbtiefe hat sich als gut erwiesen. Dieser Wert wurde für alle weiteren Experimente übernommen.

In mehreren nachfolgenden Experimenten wurde der Lagebestimmungsmodus der Sternsensoren nach jeder Bildaufnahme für etwa 20 min eingeschaltet. Damit soll eine Korrelation zwischen erfassten Sternbildern und Messdaten bestimmt werden, um die Grenzmagnitude bestimmen zu können. Während den Experimenten mit den Sternsensoren wurde sowohl Online-Telemetrie als auch erweiterte AROS-Telemetrie aufgezeichnet. Die Online-Telemetrie enthält nur das Ergebnis der Lagebestimmung in Form von Quaternionen, Informationen zum Zustand der Sensoren und Fehlercodes. Die erweiterte AROS-Telemetrie gibt Auskunft über Anzahl der detektierten Sterne, Anzahl der Übereinstimmungen im Kreismuster, Nummer des Pol-Sterns und der Nachbarsterne. Optional können die Koordinaten der Sterne im Bild sowie

die besetzten Ringe des Kreismusters angefragt werden. Die visualisierten Daten eines der Experimente sind in Abbildung 95 dargestellt.

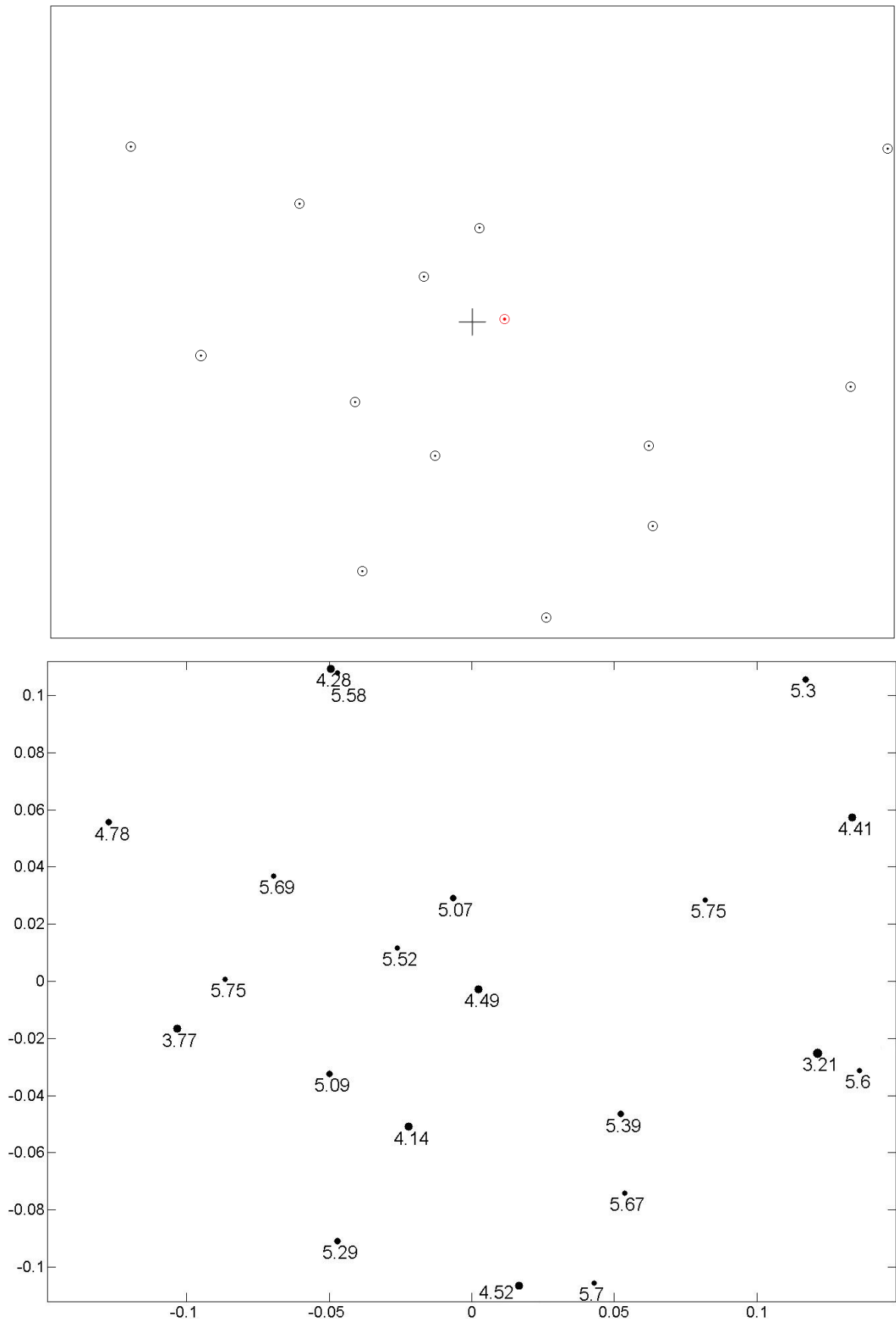


Abbildung 95. Visualisierung erweiterte Telemetrie (oben) und simulierte Aufnahme für Sterne bis zu 5,75^m (unten)

Im oberen Bild sind die vom OP1 erkannten Sterne um den in rot markierten Pol-Stern mit der Katalognummer 906 dargestellt. Der Stern wurde als Doppelstern μ 1Cyg-78Cyg HIP 107310 mit Magnitude 4,6 (4,49 im Johnson UBV photometrischen System) identifiziert. Mittels direkten Vergleichs mit einem simulierten Bild konnte die Grenzmagnitude bestimmt werden. Die Grenzmagnitude beträgt 5,75 und ist hiermit um 0,75 kleiner gegenüber den quasistatischen Aufnahmen bei den Feldtests. Diese Reduktion ist durch eine Drehbewegung und die daraus folgende Verschmierung der Sternabbildungen zu erklären. In diesem Fall wird die Lichtenergie über eine größere Fläche verteilt und kleine Sterne verschwinden unter dem Schwellenwert. Die festgestellte Reduktion der Grenzmagnitude verkleinert zwar die durchschnittliche Anzahl der Sterne im Sichtfeld, trotzdem bleiben genug Sterne für eine eindeutige Identifikation übrig. Laut durchgeführten Analysen für die Magnitude 5,75 sind immer mindestens 5 Sterne im Sichtfeld, unabhängig von der Lage des Satelliten, zu sehen.

Die ersten Versuche, die Lage zu bestimmen, zeigen, dass der Sternidentifikationsalgorithmus keine kontinuierlichen Ergebnisse liefert. Die Lücken entstehen vor allen, wenn die Drehrate des Satelliten die Grenze von ca. $0,65^\circ/\text{s}$ überschreiten. Wie bereits erwähnt wurde, verkleinert sich dadurch die Grenzmagnitude und die kleinen Sterne werden nicht mehr detektiert. Die Anzahl der Sterne im Sichtfeld sinkt unter die vordefinierte Grenze, somit ist die korrekte Identifikation nicht möglich. Außerdem wird gelegentlich ein falscher Pol-Stern bzw. Nachbarstern bestimmt. Dadurch entstehen Sprünge in den Quaternionenkurven (siehe Abbildung 96). Dies deutet auf Ungenauigkeiten in den Kalibrierungsparametern der Optik hin. Empfindliche Parameter wie Brennweite oder die Koordinaten des Bildmittelpunkts können sich nach dem Start und unter Einfluss der Weltraumbedingungen etwas verändern. Sie beeinflussen die Lage aller Sterne im Bild und dementsprechend der Kreismuster. Um die tatsächliche Brennweite genauer bestimmen zu können wurde ein separates Experiment geplant und durchgeführt. In diesem Experiment, wurde die Brennweite mit $15\mu\text{m}$ Schritten linear geändert, die relative Anzahl der richtigen Identifikationen beurteilt und ein Maximum gesucht. Es stellte sich heraus, dass sich die Brennweite um etwa $45\mu\text{m}$ verändert hat. Nachdem die Brennweite bestimmt wurde, wurde die Autokalibrierungsfunktion für mehrere Schattenphasen eingeschaltet. Durch einen innovativen Algorithmus zur Autokalibrierung passen sich die Parameter des Verzeichnungsmodells automatisch an. Das Experiment zur In-Orbit-Kalibrierung ist sehr aufwendig und zum Zeitpunkt der Niederschrift noch nicht vollständig abgeschlossen. Darüber hinaus sind die OP1-Sternsensoren nur als sekundäre Nutzlast an Bord. Deshalb haben Experimente mit den Hauptnutzlasten ASAP-Light und ADIA-Light Vorrang.

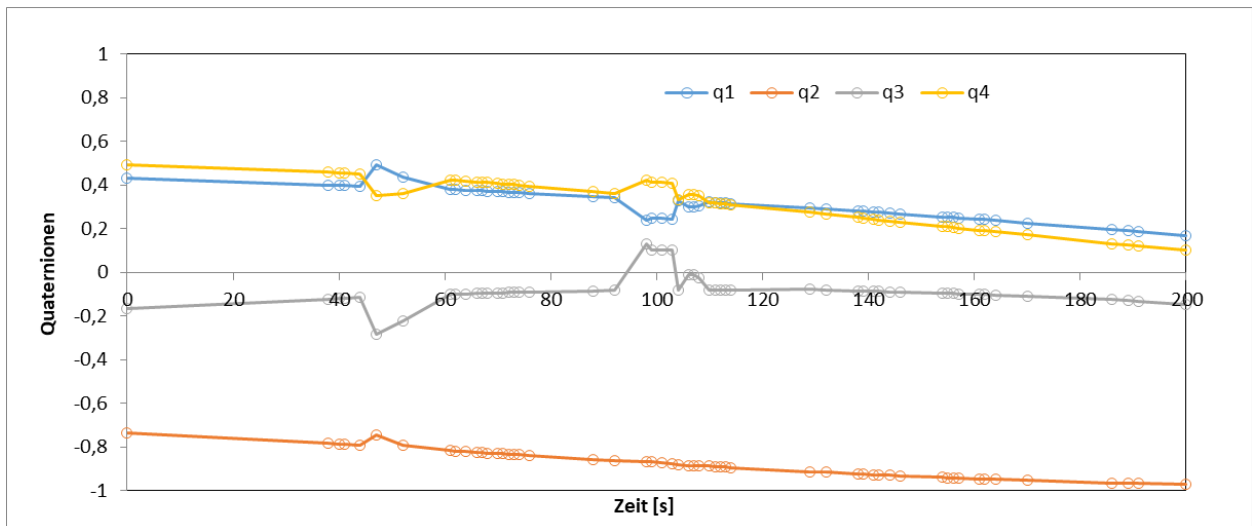


Abbildung 96. Lagedaten von einem nicht kalibrierten OPI Sensor

9 Zusammenfassung und Ausblick

Die Wahl der richtigen Systemgrößen während der Entwicklung eines Sternsensors stellt für einen Systemingenieur eine große Herausforderung dar. Komplexe Zusammenhänge im Sternsensordesign verlangen mehrere Designiterationen, bis ein gültiges Systemkonzept erstellt werden kann. Darüber hinaus werden die möglichen Designvarianten dadurch begrenzt, dass in der Entwicklung die Besonderheiten der bestimmten Satellitenklasse berücksichtigt werden sollen. Begrenzte Bauvolumen und Masse, geringe Stromaufnahme und kleine Budgets sind nur einige der Anforderungen bzw. Randbedingungen die Entwicklungen für Pico- und Nanosatelliten prägen. Außerdem geben die Missionsziele und damit verbundene Nutzlasten die Entwicklungsschwerpunkte vor. Unter Berücksichtigung all dieser Aspekte ist die Entwicklung eines Sternsensors mit der Lösung eines Multiparameter- und multikriteriellen Optimierungsproblems verbunden.

Um die Arbeit des Entwicklers in den ersten Projektphasen zu erleichtern wurde der Einsatz eines neuartigen automatischen Optimierungstools zum Design von Sternsensoren vorgeschlagen. Das entwickelte modellbasierte Softwaretool simuliert alle notwendigen Komponenten des Sternsensordesigns und verändert dabei Designvariablen, um eine Verbesserung der vordefinierten Systemeigenschaften zu erzielen. Für die gezielte Variation der Designvariablen und eine daraus resultierende Optimierung der Sternsensoren ist eine Evaluationsstrategie zuständig. Jeder Sternsensor wird als Individuum einer Population dargestellt. Die Individuen der Population durchlaufen natürliche Prozesse wie Reproduktion, Rekombination, Mutation und Selektion. Bedingt durch natürlichen Selektionsdruck, der durch die Konkurrenz zwischen den einzelnen Sternsensoren ausgedrückt wird, überleben nur die stärkeren Individuen und können sich besser über Generationen fortpflanzen. Der Fitnesswert eines Individuums beschreibt, wie gut das Individuum die vorgegebenen Ziele erreicht. Die Ziele unter Berücksichtigung der Randbedingungen werden vom Benutzer als spezielle Fitnessfunktion definiert. Am Ende der Evolution liefert das Optimierungstool ein ausgereiftes Konzept auf Systemebene. Dieses Konzept kann in nachfolgenden Entwicklungsschritten weiter bis zur konkreten technischen Lösung vertieft werden.

Die Funktionstüchtigkeit des vorgeschlagenen Tools wurde anhand einer Entwicklung von zwei Sternsensoren demonstriert. Zwei Systemkonzepte, OP1 und OP2, wurden automatisch passend zu den Anforderungen von Pico- und Nanosatelliten generiert. Für beide Konzepte wurde Hardware entwickelt, passende Algorithmen ausgewählt und erweitert, und diese in Software realisiert. Anschließend wurden die gebauten Sternsensoren einem umfangreichen Qualifizierungsprogramm samt Feldtests unterzogen. Der OP1-Sternsensor startete im Juli 2019 im Rahmen der SONATE-Mission ins All und wurde erfolgreich verifiziert. Hiermit wurde die Tragfähigkeit der Grundidee zum automatischen Systementwurf von Sternsensoren endgültig nachgewiesen. OP1 erreichte die gewünschte Zielausgaberate von 2 Hz und ein Auflösungsvermögen von $0,005^\circ$ für Pitch und Yaw sowie $0,01^\circ$ für Roll.

Das entwickelte automatische Optimierungstool erlaubt es zyklische Iterationen in der Entwurfsphase von Sternsensoren zu vermeiden. Damit verkürzt sich die Entwicklungszeit und die gewünschten Eigenschaften können gezielt optimiert werden.

Trotz massiver Hilfe durch das Optimierungstool müssen viele Entwicklungsschritte noch manuell durchgeführt werden. So berücksichtigt die Simulation der Sternensensoren z.B. keine Fertigungsungenauigkeiten bei der Herstellung der Optik. Diese werden in einer Toleranzrechnung untersucht und zeigen wie sich die Abbildungsqualität ändert, wenn verschiedene optische Größen wie Flächenkrümmung, Abstände zwischen Linsen oder radiale Verschiebung leicht variiert werden. Diese Größen können nur mit einer gewissen Fertigungstoleranz hergestellt werden. Deshalb können manche theoretisch optimalen Lösungen, die sehr empfindlich auf Änderungen reagieren, nicht hergestellt werden.

Darüber hinaus mussten manche Simulationsmodelle, wie z.B. das Bildsensormodell, vereinfacht werden, da die benötigten Informationen über interne Signalaufbereitung in den Chips nicht öffentlich sind und daher in Labor- bzw. Feldtest aufwändig ermittelt werden mussten. In der Zukunft soll ein Versuch unternommen werden, ein generisches Modell des Bildsensors zu erstellen. Außerdem soll die Optimierung mit weiteren, detaillierteren Simulationsmodellen zur Ermittlung des Stromverbrauchs und der Berechnungslaufzeit, abhängig vom Prozessortyp und für verschiedene Sternerkennungs- und Identifikationsalgorithmen, erweitert werden.

Auch das Verschmieren der Sterne bei einer Drehbewegung wurde bei der Optimierung nicht simuliert. Durch die Verteilung der Energie über eine größere Fläche im Vergleich zu quasi-statischen Aufnahmen verringert sich die Grenzmagnitude. Obwohl der ausgewählte Sternidentifikationsalgorithmus von OP1 durch entsprechende Reserven in der Magnitude unempfindlich auf diese Reduktion reagierte, sollte dieser Aspekt in Zukunft berücksichtigt werden.

Bei der Gestaltung der Evolutionsstrategie wurden in jeder neuen Generation alle Designvariablen mutiert und rekombiniert. Oft führte die Rekombination zu sehr großen Sprüngen im Suchraum. Dies betrifft vor allem die Rekombination der Linsenparameter bei der Optimierung des optischen Systems. Dadurch können Lösungen entstehen, die die Lichtstrahlen weit zerstreuen, anstatt sie zu bündeln. Um diesen Effekt vermeiden zu können, sollte es perspektivisch eine Möglichkeit geben, diesen Evolutionsoperator nur selektiv für einzelne Designvariablen anzuwenden.

Um mehrere Designziele parallel erreichen zu können, wurde als Fitnessfunktion eine gewichtete Summe verwendet. Die einzelnen Zielfunktionen wurden zusätzlich skaliert und normiert. Die Methode der gewichteten Summe war leicht zu implementieren und führt schnell zum Ergebnis. Diese Ergebnisse sind aber nur selten Pareto-optimal und können weiter verbessert werden. An dieser Stelle können andere Methoden wie z.B. NPGA effektiver arbeiten.

Abhängig von der Populationsgröße und der Anzahl der Nachkommen kann ein Optimierungsdurchlauf mehrere Tage in Anspruch nehmen. Um die Laufzeit reduzieren zu können, wurde ein Versuch unternommen, die Prozesse mit einem Standardtool von MatLab zu parallelisieren. Es wurde eine Geschwindigkeitserhöhung von 5,6-fach im Vergleich zur nicht parallelen Lösung erreicht. Um weitere Beschleunigung zu erreichen, muss das Optimie-

rungstool durch eine Portierung nach z.B. C++ oder Java zu einer eigenständigen Software werden. Damit erweitern sich die Parallelisierungsmöglichkeiten.

Für die Hardware der Sternsensoren ist es immer noch schwierig, einen passenden Bildsensor mit guter Lichtempfindlichkeit zu finden und zu beschaffen. Durch technologische Entwicklungen vor allem im Bereich der rückwärtigen Belichtung kommen neue Sensoren mit besserer Quanteneffizienz auf den Markt, die perspektivisch zu höherer Lichtempfindlichkeit der Sternsensoren beitragen können.

Im realen Betrieb von OP1 auf SONATE konnten einige neue Erkenntnisse über die Messdatenverfügbarkeit gewonnen werden. Der SONATE Satellit enthält zwei OP1-Sensoren auf gegenüberliegenden Seiten. Da der Sonnenausschlusswinkel von OP1 etwa 80° beträgt, sind die Sensoren auf der Sonnenseite der Erde häufig geblendet und liefern keine Lageinformation. Darüber hinaus verdeckt die Erde bei Satelliten im niedrigen Orbit fast die Hälfte des Sternenhimmels. Um die Verfügbarkeit der Messdaten zu erhöhen, sollten Sternsensoren entweder ein größeres FOV oder eine besseren und längeren Baffle besitzen. Dies ist für Picosatelliten aber schwer realisierbar. Alternativ können Baffle-lose Miniatur-Sternsensoren nur für die Nachtseite (eclipse) mit einem FOV größer als 90° entwickelt werden.

Mit dieser Arbeit wurde ein wichtiger Beitrag zur Automatisierung des Sternsensorentwurfs für ressourcenbeschränkte Pico- und Nanosatelliten geleistet. Mit den zusätzlichen experimentellen Erfahrungen können im Rahmen nachfolgender Projekte weitere Verbesserungen stattfinden, Auch Problematiken wie Messdatenverfügbarkeit, Kalibrierung und Optimierung der Fertigung können dann gezielt untersucht werden.

10 Literaturverzeichnis

- [AGU13] Aguirre M.R. Introduction to Space Systems: Design and Synthesis, Space technology Library, Springer, 2013
- [BAL11-1] Balagurin O., Kayal H., Wojtkowiak H., STELLA - A New Small Star Tracker for Pico and Nano Satellite, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID: IAA-B8-1201, Berlin, 2011
- [BAL11-2] Balagurin O., Kayal H., Wojtkowiak H., Test and Performance Analysis of the New Star Tracker STELLA, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Dokument- ID: 241260, Bremen, 2011
- [BAL12] Balagurin O., Kayal H., Wojtkowiak H., Validation and Qualification of a CMOS based Miniature Star Tracker for Satellite, Proceedings of 4S Symposiums, Paper-ID: 2444626, Portoroz, Slovenien, 2012
- [BAL13] Balagurin O., Fischer W., Kayal H., Wojtkowiak H., An Autonomous Imaging System with Mission Planning Capability, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Stuttgart, 2013
- [BAL14-1] Balagurin O., Fischer W., Kayal H., Wojtkowiak H., Star Sensor Solutions for Pico und Nano Satellites, Proceedings of CubeSat Workshop, 4S Symposiums, Paper-ID: 2_1540, Mallorca, Spanien, 2014
- [BAL14-2] Balagurin O., Fellingner G., Fischer W., Kayal H., Wojtkowiak H., Methods and Systems for Increasing Autonomy of Earth Observation Satellites, Proceedings of 4S Symposiums, Paper-ID: 40, Mallorca, Spanien, 2014
- [BAL15] Balagurin O., Kayal H., Schartel A., A Model-based Optimization Technique for the Development Process of Star Sensors for Pico and Nano Satellites, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Rostock, 2015
- [BAL16] Balagurin O., Kayal H., Schartel A., An Optimized Development Process for Small Star Sensor Systems, Proceedings of 4S Symposiums, Malta, Italien, 2016
- [BAL17] Balagurin O., Djebko K., Fellingner G., Kayal K., Schartel A., Schneider A., Schwarz T., Vodopivec A., Wojtkowiak H., A concept for a nanosatellite mission to demonstrate autonomous system technologies, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID: IAA-B11-0209, Berlin, 2017
- [BAL18] Balagurin O., Attitude Determination and Control, Vorlesungsskript zur Grundlagen der Raumflugmechanik, Universität Würzburg, 2018
- [BAL19] Balagurin O., Baumann T., Greiner T., Kayal H., Maurer A., Rapp T., Schwarz T., 3U satellite bus SONATE for technology demonstration of autonomous payloads, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID: IAA-B12-0902, Berlin 2019
- [BAN12] Bangert P., Optimization for industrial problems, Springer, Berlin, 2012

- [BAR13] Barschke M., Ballheimer W., Dornburg L., TechnoSat - A Nanosatellite Mission for On-Orbit Technology Demonstration, Tagungsband der Small Satellite Conference, AI-AA/Utah State University, Paper-ID: SSC13-XI-3, Utah, 2013
- [BEL99] Belanche L. A., An Application Example of the Breeder Genetic Algorithm to Function Optimisation, Buran Nr. 14, Dezember 1999
- [BOU10] Bouwmeester J., Guo J., Survey of Worldwide pico- and nanosatellite missions, distributions and subsystem technology, Acta Astronautica 67, s.854-862, 2010
- [BRI78] Brixner B. LAS lens design procedure: simple, fast, precise, versatile. Report LA-7476, Available from NTIS, PC A03/MF A01, 1978
- [BUR15] Burger W., Digitale Bildverarbeitung, 3. vollständig überarbeitete und erweiterte Auflage, Springer, Berlin-Heidelberg, 2015
- [COL04] Collette Y., Siarry P., Multiobjective Optimization: Principles and Case Studies, Springer Berlin-Heidelberg, 2004
- [COM03] Comford S.L., Feather M.S., Optimizing spacecraft design optimization engine development: progress and plans, Proceedings of IEEE Aerospace Conference, Volume 8-3681, 2003
- [CRO19] Croucher D., Increasing CMOS Camera Sensitivity Through Back-Illumination, Technical Note, Teledyne Photometrics, verfügbar unter <https://www.photometrics.com/resources/technotes/pdfs/Increasing-CMOS-Camera-Sensitivity-Through-Back-Illumination.pdf>, abgerufen am 09.10.2019
- [DAE02] Daenzer W.F., Huber F., Systems Engineering, Methodik und Praxis, Verlag Industrielle Organisation, Zürich, 11. Auflage, 2002
- [DAL05] Dalsa Teledyne, White Paper: Image Sensor Architectures for Digital Cinematography, Januar 2005
- [DER08] Dereniak E.L., Dereniak T.D., Geometrical and Trigonometrical Optics, Cambridge University Press, New York, 2008
- [DJE17] Djebko K., Schwarz T., Fellingner G., Kayal H., Puppe F., Schartel A., Vodopivec A., Wojtkowiak H., A model-based fault detection and root cause identification system for increased autonomy of small satellites, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID: IAA-B11-1305, Berlin, 2017
- [DUA09] Duarte R.O., Martins-Filho L.S., Performance Comparison of Attitude Determination Algorithms Developed to Run in a Microprocessor Environment, Proceedings of 20th International Congress of Mechanical Engineering, November 15-20, 2009
- [E2V20] E2V Flyer für EV76C454, verfügbar unter <https://imaging.teledyne-e2v.com/products/2d-cmos-image-sensors/jade/>, abgerufen am 01.02.2020

- [ECS09] ECSS-M-ST-10C, Space project management, Project planning and implementation Rev. 1, 6 March 2009
- [EIG14] Eigner M., Roubanov D., Zafirov R., Modellbasierte virtuelle Produktentwicklung, Springer Vieweg, 2014
- [ERH08] Erhardt A., Einführung in die Digitale Bildverarbeitung, 1. Auflage Vieweg + Teubner, 2008
- [EXO19] Test Service Description, Version von 15.03.2019, internes Dokument von EXOLAUNCH GmbH, kann unter info@exolaunch.com angefragt werden.
- [FED51] Feder D.P., Optical Calculations with Automatic Computing Machinery, Journal of Optical Society of America, Volume 09, 1951.
- [FED63] Feder D.P., Automatic optical design, Applied Optics, Volume 2, Issue 12, pp. 1209-1226, 1963
- [FEL13] Fellingner G., Dietrich G., Fette G., Kayal H., Puppe F., Schneider V., Wojtkowiak H., ADIA: A Novel Onboard Failure Diagnostic System for Nanosatellites. Proceedings of International Astronautical Congresses, Paper-ID: IAC-13.B4.3.10x17926, Beijing, China, 2013
- [FEL19] Fellingner G., Burger T., Djebko K., Jäger E., Test of the Autonomous Diagnostic System ADIA-Light aboard the Nanosatellite Mission SONATE, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID:IAA-B12-0908P, Berlin 2019
- [FER16] Fereyre P, Powell G., CMOS Image Sensors are entering a new age, verfügbar unter www.teledyne-e2v.com/content/uploads/2017/03/e2v_CMOS_Image_Sensors_are_Entering_a_New_Age_V4.pdf, abgerufen am 11.10.2019
- [FIS14] Fischer W., Balagurin O., Kayal H., Wojtkowiak H., Hardwarenahe Softwarelösungen für Miniaturisierte Sternsensoren, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Dokument-ID: 340071, Augsburg, 2014
- [FIS15] Fischer W., Balagurin O., Kayal H., Schartel A., Optimization of a Star Recognition Algorithm for Miniaturized Star Sensors, Proceedings of IAA Symposiums on Small Satellites for Earth Observation, Paper-ID: IAA-B10-0705P, Berlin, 2015
- [GER04] Gerdes I., Evolutionäre Algorithmen. Wiesbaden, Vieweg 2004
- [GIE81] Giese R.H., Einführung in die Astronomie, Wissenschaftliche Buchgesellschaft, XII, Darmstadt, 1981
- [GRE18] Greiner T., Balagurin O., Kayal H., Schwarz T., Optimization Tool for Mission Specific Design of Star Sensors, Proceedings of 4S Symposiums, Paper-ID:131, Sorrento, Italien, 2018
- [GRI18] Grimme C., Bossek J., Einführung in die Optimierung : Konzepte, Methoden und Anwendungen, Wiesbaden: Springer Vieweg, 2018

- [HAB99] Haberfellner R., Daenzer W.F., Systems engineering : Methodik und Praxis, 10. durchges. Auflage, Verlag Industrielle Organisation, Zürich, 1999
- [HAN14] Hanslmeier A., Einführung in Astronomie und Astrophysik, Springer-Verlag Berlin Heidelberg, 2014
- [HER17] Hernández E. A., Alonso M. A., Robust polygon recognition method with similarity invariants applied to star identification, Advances in Space Research, Volume 59, Issue 4, 2017
- [HIP20] Hipparcos-Sternkatalog, verfügbar unter <https://www.cosmos.esa.int/web/hipparcos/catalogues>, abgerufen am 01.07.2020
- [HO12] Ho K., A survey of algorithms for star identification with low-cost star trackers, ACTA Astronautica, Volume 73, 2012
- [HOR06] Hornby G.S., Globus A., Automated Antenna Design with Evolutionary Algorithms, Space 2006, Paper-ID: AIAA 2006-7242, 2006
- [JÄH12] Jähne B., Digitale Bildverarbeitung und Bildgewinnung, 7. neu bearbeitete Auflage, Springer Vieweg, Berlin, 2012
- [JAN13] Janson S., Welle R., The NASA Optical Communication and Sensors Demonstration Program: Preflight Up-date, 27th Annual AIAA/USU Conference on Small Satellites, SSC13-II-1, 2013
- [JAN15] Janson S., Welle R., The NASA Optical Communication and Sensors Demonstration Program: Preflight Up-date, 29th Annual AIAA/USU Conference on Small Satellites, SSC15-III-1, 2015
- [JAN16] Janson S., Welle R., The NASA Optical Communications and Sensor Demonstration Program: Initial Flight Results, 30th Annual AIAA/USU Conference on Small Satellites, SSC16-III-03, 2016
- [JER10] Jerram P., Burt D., Back-thinned CMOS Sensor Optimisation, Proceedings of SPIE - The International Society for Optical Engineering, Optical Components and Materials VII, Dok.-ID: 759813, 2010
- [KAI08] Kaiser A., BRITE-Constellation: Simulation of Photometric Performance, Comm. In Asteroseismology, Volume 152, 2008
- [KAR90] Karttunen H., Kröger P., Astronomie: Eine Einführung, Springer-Verlag Berlin Heidelberg, 512 s., 1990.
- [KAY09] Kayal H., Balagurin O., Wojtkowiak H., STELLA Ein neuer Sternsensor für Pico und Nanosatelliten, Tagungsband des Pico- und Nanosatelliten Workshops, Würzburg, 2009
- [KIE04] Kieran A. C. Arc-Minute Nanosatellite Attitude Control: Enabling Technology for the BRITE Stellar Photometry Mission, 18th Annual AIAA/USU Conference on Small Satellites, SSC04-V-2, 2004

- [KOL08] Kolomenkin M., Polak S., A Geometric Voting Algorithm for Star Trackers, IEEE Transaction on Aerospace and Electronic Systems, Vol. 44, No. 2, April 2008
- [KON06] Konak A., Coit D.W., Smith A.E., Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering & System Safety, Volume 91, Issue 9, 2006
- [KOS03] Kossiakoff A, Sweet W. N., System Engineering Principles and Practice, Second Edition, John Wiley & Sons, New Jersey, 2003
- [KOS91] Kosik J. C., Star pattern identification aboard an inertially stabilized aircraft, Journal of Guidance, Control and Dynamics, Volume 24, No. 2, 1991
- [LAM05] Lamy Au Rousseau G., Bostel J., Mazari B., Star recognition algorithm for APS star tracker: oriented triangles, Aerospace and Electronic Systems Magazine, Volume 20, Issue 2, 2005
- [LEE03] Lee H., Oh C.S., Bang H, Modified grid algorithm for star pattern identification by using star trackers, International Conference on Recent Advances in Space Technologies, Dok-ID.: 8160281, 2003
- [LI03] Li Chunyan, Li Ke, Star pattern recognition method based on neural network, Chinese Science Bulletin, Volume 48, No. 18, 2003
- [LI10] Li L., Wang Q.H., Xu X.O., Li D.H., Two-step method for lens system design, Optics Express, Volume 18, Issue 12, 2010
- [LIE02] Liebe C.C., Accuracy Performance of Star Trackers –A Tutorial, IEEE Transactions on Aerospace and Electronic systems, Volme 38, No. 2, April 2002
- [LIE93] Liebe C. C., Pattern Recognition of Star Constellations for Spacecraft Applications, IEEE AES Systems Magazine, January 1993.
- [LIE95] Liebe C. C., Star Trackers for Attitude Determination, IEEE AES Systems Magazine, June, 1995
- [LIN02] Lin Y., Deng Z., Star-Sensor-Based Predictive Kalman Filter for Satellite Attitude Estimation, Science in China, Volume 45, No. 3, 2002
- [LIN97] Lindsey C. S., Lindblad T., Method for star identification using neural networks, SPIE - The International Society for Optical Engineering, 1997
- [LIN17] Ling P.D., Advanced Geometrical Optics, Springer Science+Business Media Singapore, 2017
- [LIT01] Litwiller D., CCD vs. CMOS: Facts and Fiction, Photonics Spectra, Laurin Publishing Co.Inc., 2001
- [LIT05] Litwiller D, CMOS vs. CCD: Maturing Technologies, Maturing Markets, Photonics Spectra, Laurin Publishing Co.Inc., 2005

- [LIT13] Litfin G., Technische Optik in der Praxis, Springer-Verlag, 2013
- [LUO15] Luo L., Xu L., Zhang H., An Autonomous Star Identification Algorithm Based on One-Dimensional Vector Pattern for Star Sensors, *Sensors*, Volume 15, Issue 7, Basel, 2015
- [MAL04] Malacara D., Malacara Z., Handbook of Optical Design, Second Edition, Marcel Dekker Inc., New York, 2004
- [MAR00] Markley F.L., Mortari D., Quaternion Attitude Estimation Using Vector Observations, *The Journal of the Astronautical Sciences*, Volume 48, No. 2 und 3, 2000
- [MAR04] Marler R.T., Arora J.S., Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization*, Volume 26, Issue 6, 2004
- [MAR14] Marklay F. L., Fundamentals of spacecraft attitude determination and control, Springer, New York, 2014
- [MAR88] Markley F.L., Attitude Determination Using Vector Observations and the Singular Value Decomposition, *The Journal of the Astronautical Sciences*, Volume 36, Issue 3, 1988
- [MAR93] Markley F.L., Attitude Determination Using Vector Observations, A Fast Optimal Matrix Algorithm, *The Journal of the Astronautical Sciences*, Volume 41, Issue 2, 1993
- [MIL62] MIL-HDBK-141, Optical Design, Military Standardization Handbook, 1962
- [MOR00] Mortari D., Neta B., k-Vector Range Searching Techniques, *Advances in Astronautical Sciences*, Volume 105, 2000
- [MOR04] Mortari D., Samaan A.M., The Pyramid Star Identification Technique, *Journal of the Institute of Navigation*, Volume 51, Issue 3, 2004
- [MOS98] Mosher T., Spacecraft design using a genetic algorithm optimization approach, *IEEE Aerospace Conference Proceedings*, 1998
- [NA06] Na M., Jia P., A survey of all-sky autonomous star identification algorithms, *IEEE, International Symposium on Systems and Control in Aerospace and Astronautics*, Paper-ID: 9005061, 2006
- [NAK05] Nakamura J., Image Sensors and Signal Processing for Digital Still Cameras, Taylor & Francis Group, 2005
- [NAN19] Nanosatellitendatenbank, Abbildungen URL: <https://www.nanosats.eu/#figures>, Abruf am 02.09.2019
- [NAT16] National Academies of Sciences, Engineering and Medicine, Achieving Science with CubeSats: Thinking Inside the Box, National Academies Press, Washington, 2016
- [NAS17] NASA CubeSat Launch Initiative, Cube Sat 101: Basic Concepts and Processes for First-Time CubeSat Developers, 2017, verfügbar unter <https://www.cubesat.org/cubesatinfo>, abgerufen am 03.03.2021

- [NGA05] Ngatchou P., Zarei A., El-Sharkawi M.A., Pareto Multi Objective Optimization, Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems, 2005
- [NIS11] Nischwitz A., Computergrafik und Bildverarbeitung, Band 2: Bildverarbeitung, 3. neu bearbeitete Auflage, Vieweg + Teubner, Wiesbaden, 2011
- [NIS97] Nissen N., Einführung in Evolutionäre Algorithmen: Optimieren nach Vorbild der Evolution, Wiesbaden, Vieweg, 1997
- [ONO00] Ono I., Kobayashi S., Yochida k., Optimal lens design by real-coded genetic algorithms using UNDX, Computer Methods in Applied Mechanics and Engineering, Volumen 186, Issue 2-4, 2000
- [ONS19-1] On Semiconductor Datenblatt für AR130, verfügbar unter <https://www.onsemi.com/support/design-resources/datasheets?category=101682>, abgerufen am 03.03.2019
- [ONS19-2] On Semiconductor Datenblatt für ASX340CS, verfügbar unter <https://www.onsemi.com/support/design-resources/datasheets?category=101682>, abgerufen am 03.03.2019
- [OSH85] O'Shea D. C., Elements of modern optical design, 1. Auflage, Wiley-Interscience, 1985
- [PAB16] Pablo H., Whittaker G.N., Popowicz A., et al., The BRITE Constellation Nanosatellite Mission: Testing, Commissioning, and Operations, Astronomical Society of the Pacific 128(970), 2016
- [PAD97-1] Padgett C., A Grid Algorithm for Autonomous Star Identification, IEEE Transactions on Aerospace and Electronic Systems, Volume 33, Issue 1, 1997
- [PAD97-2] Padgett C., Evaluation of Star Identification Techniques, Journal of Guidance, Control and Dynamics, Volume 20, Issue 2, 1997
- [PED08] Pedrotti F., Pedrotti L., Bausch W., Schmidt H., Optik für Ingenieure: Grundlagen, 4. Auflage Springer-Verlag Berlin Heidelberg, 2008
- [POH99] Pohlheim H., Evolutionäre Algorithmen – Verfahren, Operatoren, Hinweise aus der Praxis. Springer-Verlag, 1999
- [QUI96] Quine B. M., Durrant-Whyte H. F., A Fast Autonomous Star-Acquisition Algorithm for Spacecraft, Control Engineering Practice, Volume 4, Issue 12, 1996
- [RAP18] Rapp T., Balagurin O., Baumann T., Kayal H., Krainovic A., Schwarz T., Wojtkowiak H., Preparing SONATE for Autonomous Control Through ASAP, Proceedings of International Astronautical Congresses, Paper-ID:IAC-17.B4.3.13x45060, Bremen, 2018
- [RIC10] Ricolfe C. V., Lens Distortion Models Evaluation, Applied Optics, Vol. 49, Issue 30, s. 5914-5928, Oktober 2010

- [RIC15] Richerdson G., Schmitt K., Coverrt M., Rogers C., Small Satellite Trends 2009-2013. AIAA/USU Conference on Small Satellites, Dok.-ID: SSC15-VII-3, 2015
- [ROB05] Roberts P.J., Walker R.A., Application of a Counter Propagation Neural Network for Star Identification, AIAA Guidance, Navigation, and Control Conference and Exhibit, Paper-ID AIAA 2005-6469, San Francisco, California, 2005
- [ROE89] Roelof W. H. van Bezooijen, A Star Pattern Recognition Algorithm for Autonomous Attitude Determination, IFAC Automatic Control in Aerospace, Volume 22, Issue 7, 1989
- [ROE92] Roelof W. H. van Bezooijen, Autonomous Star Tracker Development, IFAC Automatic Control in Aerospace, Volume 25, Issue 22, 1992
- [ROE94] Roelof W. H. van Bezooijen, True-sky demonstration of an autonomous star tracker, SPIE International Symposium on Optical Engineering and Photonics in Aerospace, Volume 2221/157, 1994
- [ROW18] Rowen D., Janson S., The NASA Optical Communications and Sensor Demonstration Program: Proximity Operations, 32nd Annual AIAA/USU Conference on Small Satellites, SSC18-I-05, 2018
- [SAM05] Samaan A.M., Mortari D., Recursive Mode Star Identification Algorithms, IEEE Transaction on Aerospace and Electronic Systems, Volume 41, Issue 4, October 2005
- [SAM06] Samaan A.M., Mortari D., Nondimensional Star Identification for Un-Calibrated Star Cameras, The Journal of the Astronautical Sciences, Volume 54, Issue 1, 2006
- [SCH10] Schwarzenberg-Czerny A., Weiss W.W., Moffat A.F.J, Et al., The BRITE Nano-Satellite Constellation Mission, 38th COSPAR Scientific Assembly, 2010
- [SCH15] Schwarz T., Balagurin O., Kayal H., Schartel A., PicoStar - Prototyping of a Star Sensor for Pico-Satellites, Proceedings of Pico- und Nanosatelliten Workshops, Würzburg, 2015
- [SCH16-1] Schartel A, Balagurin O., Kayal H., Schwarz T., Ein Micro-Display basierter Sternsimulator für Indoor-Tests miniaturisierter Sternsensoren, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Braunschweig, 2016
- [SCH16-2] Schwarz T., Balagurin O., Djebko K., Fellingner G., Kayal H., Puppe F., Schartel A., Vodopivec A. und Wojtkowiak H., SONATE Nanosatelliten-Mission zur Erprobung von hochautonomen Nutzlasten, Tagungsband des Deutscher Luft- und Raumfahrtkongresses, Braunschweig, 2016
- [SCH18-1] Scholz M., Die Physik der Sterne, Springer Spektrum, 2018
- [SCH18-2] Schwarz T., Balagurin O., Baumann T., Greiner T., Kayal H., Maurer A., Rapp T., SONATE - 3U nano Satellite Mission for Highly autonomous payloads, Proceedings of 4S Symposiums, Paper-ID: 98, Sorrento, Italien, 2018

- [SCH19-1] Schott Abbe- Diagramm., verfügbar unter: http://www.schott.com/advanced_optics/german/download/index.html, abgerufen am 19.03.2019
- [SCH19-2] Schott, Positive List of Optical Glass, Verfügbar unter: http://www.schott.com/advanced_optics/german/download/index.html, abgerufen am 19.03.2019
- [SCH87] Schröder G., Technische Optik: Grundlagen und Anwendungen, 6. Auflage, Würzburg, Vogel, 1987
- [SCH94] Schöneburg E., Heinzmann F., Feddersen S., Genetische Algorithmen und Evolutionsstrategien: Eine Einführung in Theorie und Praxis der simulierten Evolution. 1. Auflage, Addison-Wesley GmbH, 1994.
- [SHU81] Shuster M.D., Oh S.D., Three-Axis Attitude Determination from Vector Observations, Journal of Guidance, Control and Dynamics, Volume 4, No. 1, Paper-ID: AIAA 81-4003, 1981
- [SIL06] Silani E., Lovera M., Star Identification Algorithms: Novel Approach & Comparison Study, IEEE Transactions on Aerospace and Electronic Systems, Volume 42, Issue 4, 2006
- [SIO17] Sioshansi R., Conejo A. J., Optimization in engineering: models and algorithms, Springer, 2017
- [SLJ69] Sljusarev G.G, „Методы расчота ортических систем“, Maschinostroenie, Leningrad, 1969
- [SMI00] Smith W.J., Modern optical engineering, 3rd Edition, McGraw-Hill, 2000
- [SPR09] Spratling B.B., Mortari D., A Survey on Star Identification Algorithms, Algorithms 2 (1), 2009
- [STA65] Stavroudis N., Loyd E., Spot Diagrams for the Prediction of Lens Performance From Design Data. U.S. Department of Commerce, NBS Monograph 93, 1965
- [STE94] Steyn W.H., Full Satellite State Determination from Vector Observations, IFAC Proceedings of Symposium on Automatic Control in Aerospace, Volume 27, Issue 13, 1994
- [SWA19] Swartwout M. A., CubeSat Mission Success: Are We Getting Better?, presented at the 2019 CubeSat Developers Workshop, San Luis Obispo, CA, 23 April 2019
- [THI05] Thibault S., Gagne C., Beaulieu J., Evolutionary algorithms applied to lens design: case study and analysis, Optical Design and Engineering II, SPIE Proceedings, Volume 5962, 2005
- [VIL19] Villela T., Costa C.A. Towards the Thousandth CubeSat: A Statistical Overview, International Journal of Aerospace Engineering, Volume 2019, Article ID 5063145, 2019
- [WAH65] Wahba G., Problem 65–1: A Least Squares Estimate of Spacecraft Attitude, SIAM Review, Volume 7, Issue 3, 1965

- [WEI13] Weiss W. W., *BRITE-Constellation: Nanosatellites for Precision Photometry of Bright Stars*, Publications of the Astronomical Society of the Pacific, 126, 2013
- [WEI15] Weicker K., *Evolutionäre Algorithmen*, 3. Auflage, Springer Fachmedien, Wiesbaden, 2015
- [WET2020] Weston S., *State of the Art of Small Spacecraft Technology. Update 2020*, verfügbar unter: www.nasa.gov/smallsat-institute/sst-soa, abgerufen am 03.03.2021
- [WOJ11] Wojtkowiak H., Balagurin O., Kayal H., *STELLA: A Novel APS Star Tracker for Pico- and Nano Satellites*, Proceedings of International Astronautical Congresses, IAC-11.C1.6.7x10267, Kapstadt, 2011
- [WOJ13-1] Wojtkowiak H., Balagurin O., Fellingner G., Kayal H., *ASAP: Increasing the autonomy of small satellites*, Proceedings of IAA Symposia on Small Satellites for Earth Observation, Paper-ID:IAA-B9-1109P, Berlin, 2013
- [WOJ13-2] Wojtkowiak H., Balagurin O., Fellingner G., Kayal H., *ASAP: Autonomy Through On-Board Planning*, 6th International Conference on Recent Advances in Space Technologies (RAST), Paper-ID: 10.1109/RAST.2013.6581235, Istanbul, Türkei, 2013
- [WOJ14] Wojtkowiak H., Balagurin O., Fellingner G., Fischer W., Borja G. Kayal H., *ASAP: Autonomous Onboard Missing Planing*, Proceedings of International Astronautical Congresses, Paper-ID:IAAC-14-B6.2.2, Toronto, Canada, 2014
- [WOJ17] Wojtkowiak H., Balagurin O., Djebko K., Gerhard F., Greiner T., Kayal H., Schneider A., Schwarz T., *Autonomous Mission Operation Onboard Nano-Satellite SONATE*, Proceedings of International Astronautical Congresses, Paper-ID:IAAC-17-B6.2.2.37561, Adelaide, 2017
- [WÜL04] Wüller D., *CCD kontra CMOS: Die verschiedenen Sensortypen von Digitalkameras*, ColorFoto, Ausgabe 8, 2004
- [ZHA17] Zhang G., *Star Identification: Methods, Techniques and Algorithms*, Springer-Verlag, 2017
- [ZIM08] Zimmerman W. B., *Optics*, Indiana University South Bend, 2008, Verfügbar unter: <http://www.learnoptics.com>, abgerufen am 13.08.2018

11 Anhang

Anhang A: Begriffe der Bildsensortechnologie

Beim Vergleich der Detektortechnologien wurden verschiedene spezifische Begriffe verwendet. Diese sollen nun genauer erläutert werden.

Quanteneffizienz (QE)

Das Verhältnis zwischen der Anzahl an Photonen, die die lichtempfindliche Fläche des Detektors erreicht haben, und der Anzahl an Photoelektronen, die im Detektor generiert wurden, wird Quanteneffizienz genannt. Sie ist wellenlängenabhängig und je nach Pixeltechnologie und Pixelarchitektur unterschiedlich ausgeprägt. Beispielkurven für die QE sind in Anhang F zu finden.

Füllfaktor (FF)

Nicht die ganze Fläche eines Pixels ist lichtempfindlich. Außer der Photodiode besteht ein Pixel aus anderen Peripheriekomponenten, meistens Transistoren, die Licht reflektieren oder absorbieren. Der Füllfaktor beschreibt das Verhältnis zwischen der lichtsensitiven Fläche des Pixels und dessen Gesamtfläche. Normalerweise enthält die QE – Kurve Angaben zum FF.

Umwandlungskoeffizient (Conversion Gain)

Der Umwandlungskoeffizient zeigt an, wie sich die Ausgangsspannung eines Pixels mit Änderung seiner Ladung verändert. Dieser Koeffizient ist umgekehrt proportional zur "Full-Well"-Kapazität - der maximalen Kapazität des Pixels, ausgedrückt in Elektronen. Ein Pixel verhält sich bei der Belichtung wie ein Kondensator. Somit ist die Spannung gleich Ladungsmenge geteilt durch Kapazität.

Empfindlichkeit (Sensitivity / Responsivity)

Die Empfindlichkeit des Sensors wird oft in den photometrischen Einheiten $\frac{V}{lux \cdot s}$ bzw. $\frac{LSB}{lux \cdot s}$ oder in der radiometrischen Einheit $\frac{A}{W}$ ausgedrückt. Sie beschreibt das Verhältnis des elektrischen bzw. digitalen Ausgangssignales zur Beleuchtungsstärke am Eingang des Sensors für eine vordefinierte Integrationszeit. Die Empfindlichkeit fasst alle fotoelektrischen Abläufe in einem Bildsensor zusammen. Die Messung der Empfindlichkeit erfolgt anhand einer Referenzlichtquelle bei einer bestimmten Wellenlänge. Eine typische Kurve ist in Abbildung 97 zu sehen. Die Angaben zur Empfindlichkeit können oft direkt aus den Datenblättern der Detektoren entnommen werden. Die „Responsivity“ bezieht sich auf das Gesamtspektrum des Lichtes und ist ähnlich zur QE.

Signal-Rausch-Verhältnis (SNR)

Anhand der Empfindlichkeit allein ist es nicht möglich, die Helligkeit des schwächsten Objekts, das mit dem gegebenen Sensor noch aufgenommen werden kann, zu ermitteln. Eine weitere dafür benötigte Größe ist das SNR. Es gibt an, wie groß der Rauschanteil in Bezug

auf das Nutzsignal ist. Unter Rauschen versteht man das Gesamtrauschen aus Dunkelstrom- und Auslese- sowie Photonenstromrauschen. Je nach Inputsignal und Bedingungen dominiert der eine oder andere Rauschtyp.

$$SNR = 20 \log \left(\frac{N_{sig}}{N_{noise}} \right)$$

In der Sternsensoranwendung dominieren Dunkelstrom- und Ausleserauschen. Die Abbildung eines noch detektierbaren Sterns muss über dem Rauschniveau liegen.

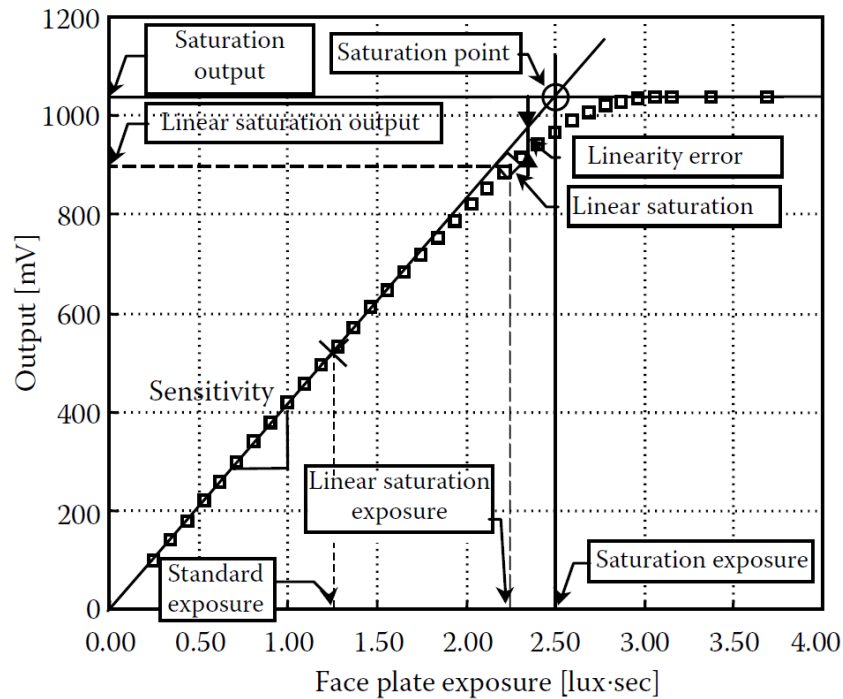
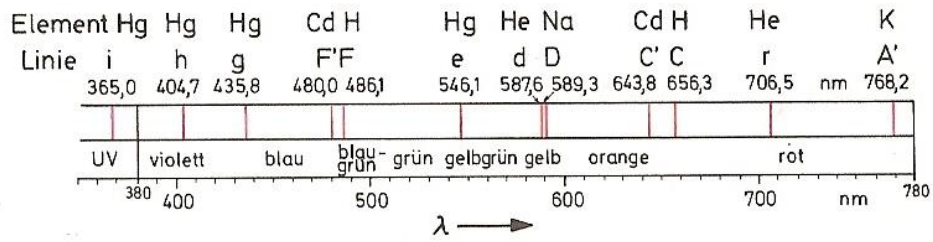
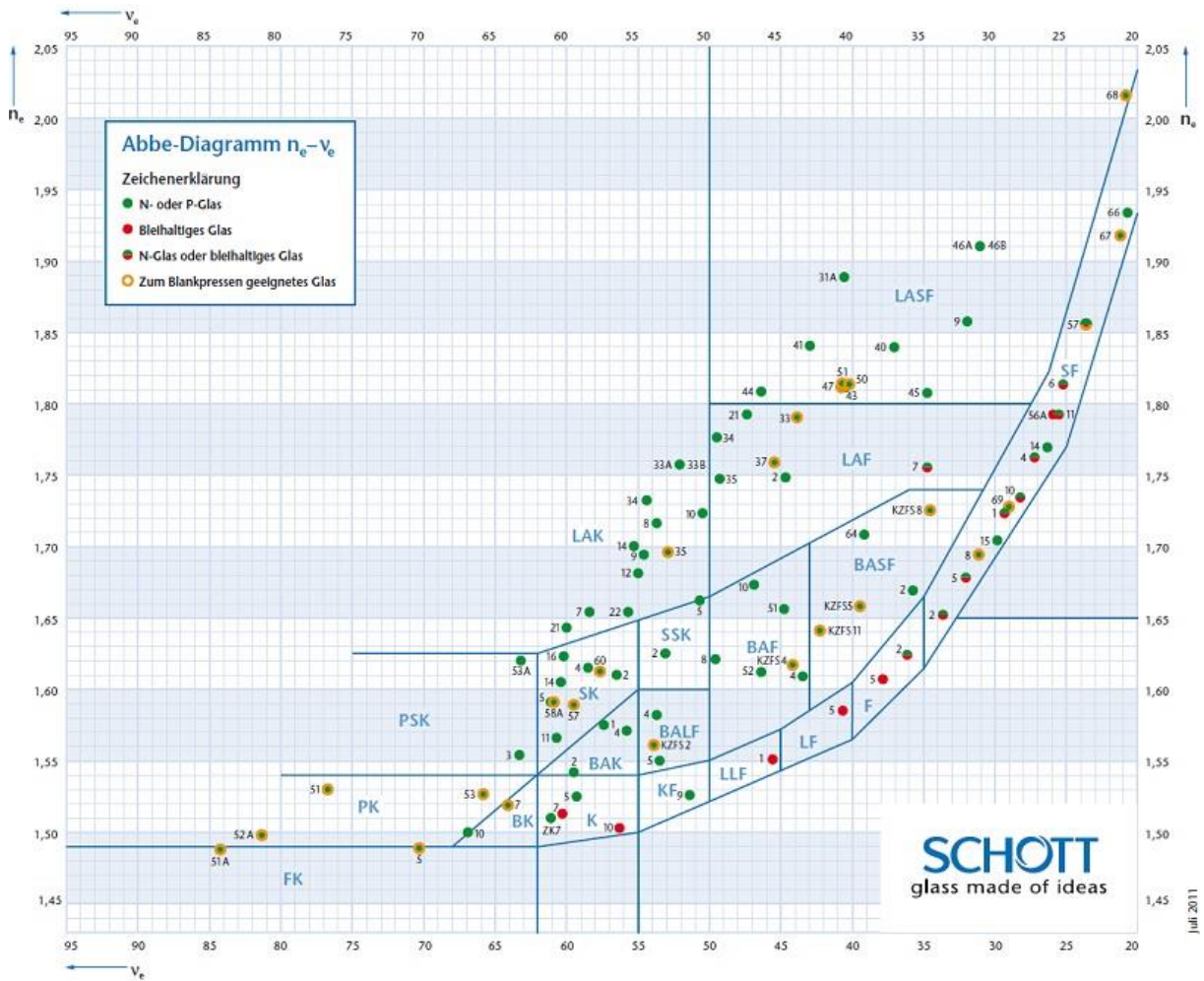


Abbildung 97. Empfindlichkeit eines Detektors [NAK05]

Anhang B: Spektrale Linien und Spektralbereiche [SCH87].



Anhang C: Abbe-Diagramm [SCH19-1]



Anhang D: Vorzeichensystem für die Strahlengänge [OSH85]

1. Bis zur ersten Reflexion verlaufen Lichtstrahlen durch alle refraktiven Flächen von links nach rechts. Alle nach rechts gemessenen Strecken und die dazugehörigen Brechzahlen sind positiv.
2. Alle Brennweiten von konvergierenden optischen Flächen sind positiv, von divergierenden sind sie negativ.
3. Durchgangshöhen y von Strahlen oberhalb der optischen Achse sind positiv, unterhalb sind sie negativ.
4. Die Abstände t rechts vom Scheitelpunkt der optischen Fläche sind positiv, links sind sie negativ.
5. Der Krümmungsradius R der Fläche ist positiv, wenn ihr Krümmungsmittelpunkt rechts von Scheitelpunkt liegt und negativ, wenn links davon.
6. Der Winkel i des Strahls im Bezug zur Normalen der Fläche oder u zur optischen Achse ist positiv, wenn der Strahl bei der Drehung im Uhrzeigersinn mit dem Bezugselement zusammenfällt. Andernfalls ist der Winkel negativ.
7. Nach einer Reflexion ändern sich die Vorzeichen der Brechzahlen. Für alle Abstände wird Regel 4 verwendet: Abstände zwischen den Flächen und entlang des Strahles sind negativ. Alle anderen Größen bleiben unverändert.

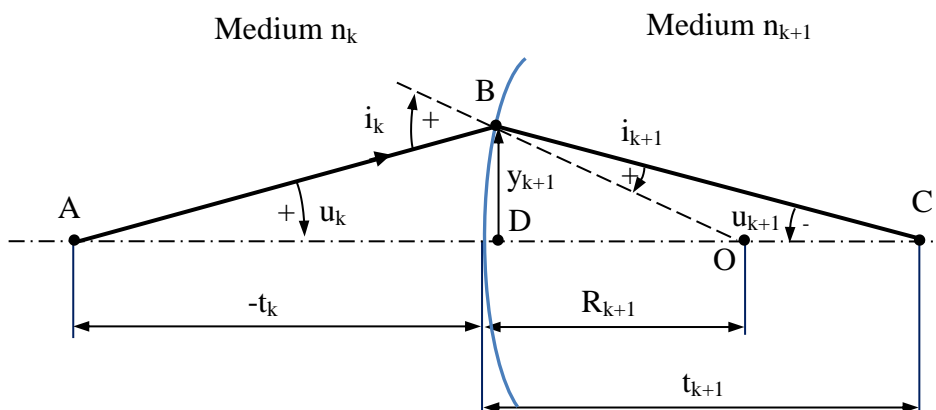


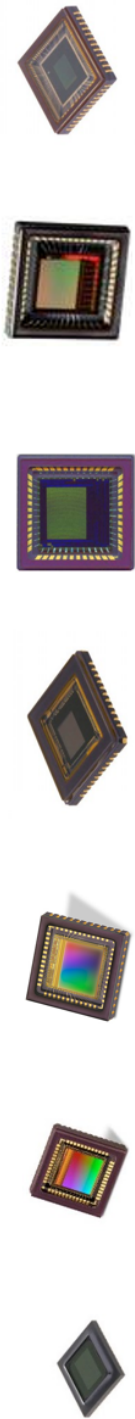
Abbildung 98. Vorzeichenregeln [OSH85]

Bemerkung: Eine vollständige Umkehrung der Vorzeichen in den oben aufgelisteten Regeln führt auch zum richtigen Ergebnis. Dies muss während der gesamten Berechnung aber konsistent bleiben.

Anhang E: Liste der bevorzugten Gläser für die Berechnung der Brechzahlen nach Sellmeier [SCH19-2]

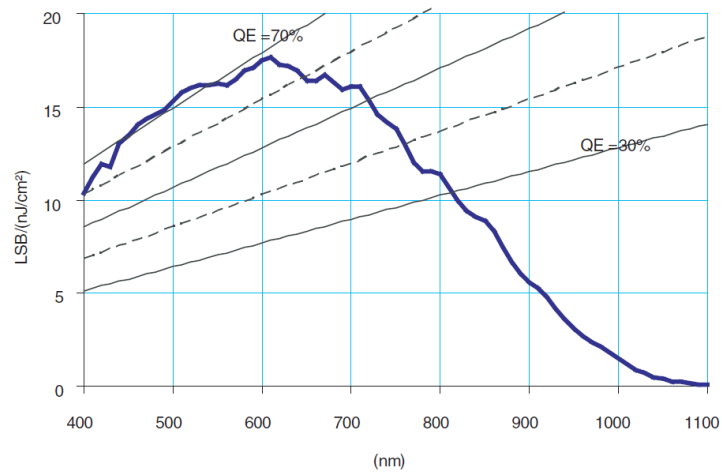
	B1	B2	B3	C1	C2	C3	Glass
1	1,33222450E+00	2,89241610E-01	1,15161734E+00	8,40298480E-03	3,44239720E-02	8,84310532E+01	N-KZFS11
2	1,75836958E+00	3,13537785E-01	1,18925231E+00	8,72810026E-03	2,93020832E-02	8,51780644E+01	N-LAF34
3	1,96485075E+00	4,75231259E-01	1,48360109E+00	9,82060155E-03	3,44713438E-02	1,10739863E+02	N-LASF31A
4	1,72878017E+00	1,69257825E-01	1,19386956E+00	8,86014635E-03	3,63416509E-02	8,29009069E+01	N-LAK10
5	1,46231905E+00	3,44399589E-01	1,15508372E+00	7,24270156E-03	2,43353131E-02	8,54686868E+01	N-LAK9
6	1,14229781E+00	5,35138441E-01	1,04088385E+00	5,85778594E-03	1,98546147E-02	1,00834017E+02	N-LAK22
7	1,80984227E+00	1,57295550E-01	1,09300370E+00	1,01711622E-02	4,42431765E-02	1,00687748E+02	N-LAF2
8	1,23679889E+00	4,45051837E-01	1,01745888E+00	6,10105538E-03	2,01388334E-02	9,06380380E+01	N-LAK7
9	1,58514950E+00	1,43559385E-01	1,08521269E+00	9,26681282E-03	4,24489805E-02	1,05613573E+02	N-BAF10
10	1,43060270E+00	1,53150554E-01	1,01390904E+00	8,23982975E-03	3,33736841E-02	1,06870822E+02	N-SSK2
11	1,19286778E+00	8,93346571E-02	9,20819805E-01	8,39154696E-03	4,04010786E-02	1,12572446E+02	N-KF9
12	1,51503623E+00	1,53621958E-01	1,15427909E+00	9,42734715E-03	4,30826500E-02	1,24889868E+02	N-BAF51
13	1,31004128E+00	1,42038259E-01	9,64929351E-01	7,96596450E-03	3,30672072E-02	1,09197320E+02	N-BALF4
14	1,34317774E+00	2,41144399E-01	9,94317969E-01	7,04687339E-03	2,29005000E-02	9,27508526E+01	N-SK16
15	1,28834642E+00	1,32817724E-01	9,45395373E-01	7,79980626E-03	3,15631177E-02	1,05965875E+02	N-BAK4
16	1,21640125E+00	1,33664540E-01	8,83399468E-01	8,57807248E-03	4,20143003E-02	1,07593060E+02	LLF1
17	1,03961212E+00	2,31792344E-01	1,01046945E+00	6,00069867E-03	2,00179144E-02	1,03560653E+02	N-BK7
18	8,88308131E-01	3,28964475E-01	9,84610769E-01	5,16900822E-03	1,61190045E-02	9,97575331E+01	N-BK10
19	1,28189012E+00	2,57738258E-01	9,68186040E-01	7,27191640E-03	2,42823527E-02	1,10377773E+02	N-SK2
20	9,91463823E-01	4,95982121E-01	9,87393925E-01	5,22730467E-03	1,72733646E-02	9,83594579E+01	N-SK5
21	1,59222659E+00	1,03520774E-01	1,05174016E+00	9,20284626E-03	4,23530072E-02	1,06927374E+02	N-SSK5
22	1,65554268E+00	1,71319770E-01	1,33664448E+00	1,04485644E-02	4,99394756E-02	1,18961472E+02	N-BASF64
23	8,87272110E-01	4,89592425E-01	1,04865296E+00	4,69824067E-03	1,61818463E-02	1,04374975E+02	N-PSK3
24	1,28385965E+00	7,19300942E-02	1,05048927E+00	8,25815975E-03	4,41920027E-02	1,07097324E+02	N-BALF5
25	1,17963631E+00	2,29817295E-01	9,35789652E-01	6,80282081E-03	2,19737205E-02	1,01513232E+02	N-SK11
26	1,31044630E+00	1,96034260E-01	9,66129770E-01	9,58633048E-03	4,57627627E-02	1,15011883E+02	F5
27	1,28035628E+00	1,63505973E-01	8,93930112E-01	9,29854416E-03	4,49135769E-02	1,10493685E+02	LF5
28	1,12735550E+00	1,24412303E-01	8,27100531E-01	7,20341707E-03	2,69835916E-02	1,00384588E+02	K7
29	1,08511833E+00	1,99562005E-01	9,30511663E-01	6,61099503E-03	2,41108660E-02	1,11982777E+02	N-K5
30	1,34533359E+00	2,09073176E-01	9,37357162E-01	9,97743871E-03	4,70450767E-02	1,11886764E+02	F2
31	1,12365662E+00	3,09276848E-01	8,81511957E-01	6,44742752E-03	2,22284402E-02	1,07297751E+02	N-BAK1
32	1,44857867E+00	1,17965926E-01	1,06937528E+00	8,69310149E-03	4,21566593E-02	1,11300666E+02	N-SSK8
33	1,01662154E+00	3,19903051E-01	9,37232995E-01	5,92383763E-03	2,03828415E-02	1,13118417E+02	N-BAK2
34	1,40301821E+00	2,31767504E-01	9,39056586E-01	1,05795466E-02	4,93226978E-02	1,12405955E+02	SF2
35	1,46141885E+00	2,47713019E-01	9,4995832E-01	1,11826126E-02	5,08594669E-02	1,12041888E+02	SF5
36	1,43903433E+00	9,67046052E-02	1,09875818E+00	9,07800128E-03	5,08212080E-02	1,05691856E+02	N-BAF52
37	1,42056328E+00	1,02721269E-01	1,14380976E+00	9,42015382E-03	5,31087291E-02	1,10278856E+02	N-BAF4
38	2,45505861E+00	4,53006077E-01	2,38513080E+00	1,35670404E-02	5,45803020E-02	1,67904715E+02	LASF35
39	8,44309338E-01	3,44147824E-01	9,10790213E-01	4,75111955E-03	1,49814849E-02	9,78600293E+01	N-FK5
40	2,00029547E+00	2,98926886E-01	1,80691843E+00	1,21426017E-02	5,38736236E-02	1,56530829E+02	N-LASF9
41	1,74028764E+00	2,26710554E-01	1,32525548E+00	1,07925580E-02	5,38626639E-02	1,06268665E+02	N-LAF7
42	2,16701566E+00	3,19812761E-01	1,66004486E+00	1,23595524E-02	5,60610282E-02	1,07047718E+02	N-LASF46A
43	1,55912923E+00	2,84246288E-01	9,68842926E-01	1,21481001E-02	5,34549042E-02	1,12174809E+02	SF1
44	1,38121836E+00	1,96745645E-01	8,86089205E-01	7,06416337E-03	2,33251345E-02	9,74847345E+01	N-PSK53A
45	1,53652081E+00	1,56971102E-01	1,30196815E+00	1,08435729E-02	5,62278762E-02	1,31339700E+02	N-BASF2
46	1,61957826E+00	3,39493189E-01	1,02566931E+00	1,25502104E-02	5,44559822E-02	1,17652222E+02	SF4
47	1,57055634E+00	2,18987094E-01	1,50824017E+00	1,16507014E-02	5,97856897E-02	1,32709339E+02	N-SF15
48	1,69022361E+00	2,88870052E-01	1,70451870E+00	1,30512113E-02	6,13691880E-02	1,49517689E+02	N-SF14
49	1,15610775E+00	1,53229344E-01	7,85618966E-01	5,85597402E-03	1,94072416E-02	1,40537046E+02	N-PK51
50	9,71247817E-01	2,16901417E-01	9,04651666E-01	4,72301995E-03	1,53575612E-02	1,68681330E+02	N-FK51A
51	1,73848403E+00	3,11168974E-01	1,17490871E+00	1,36068604E-02	6,15960463E-02	1,21922711E+02	SF11
52	1,81651371E+00	4,28893641E-01	1,07186278E+00	1,43704198E-02	5,92801172E-02	1,21419942E+02	SF57
53	1,72448482E+00	3,90104889E-01	1,04572858E+00	1,34871947E-02	5,69318095E-02	1,18557185E+02	SF6
54	1,15687082E+00	6,42625444E-02	8,72376139E-01	8,09424251E-03	3,86051284E-02	1,04747730E+02	K10
55	1,32993741E+00	2,28542996E-01	9,88465211E-01	7,16874107E-03	2,46455892E-02	1,00886364E+02	N-SK4

Anhang F: Auswahl der Bildsensoren

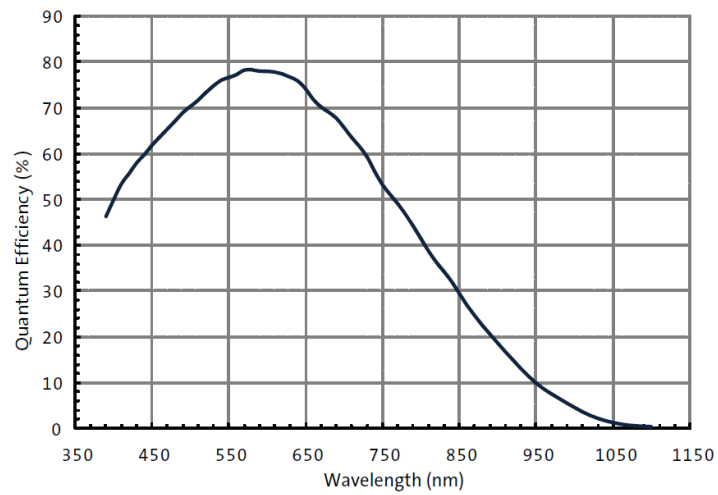


Parameters	ASX340CS, Aptina	EV76C454, E2V	EV76C661, E2V	IMX035LLR-C, Sony	Lupa-300, ON	Python 1.3, ON	AR130, Aptina
Resolution	0,4MP	0,6MP	1,3MP	1,31MP	0,3MP	1,3MP	1.2MP
Optical Format	1/4"	1/2,9"	1/1,8"	1/3"	1/2"	1/2"	1/3"
Active Array	728 x 560	838 x 640	1280 x 1024	1280 x 1024	640x480	1280x1024	1280x960
Pixel Size	5,6µm	5,5µm	5,3µm	3,6µm	9,9µm	4,8µm	3,75µm
Chroma	10-Bit, RGB	8-Bit, Mono	8-Bit, Mono	12-Bit, Mono	10-Bit, Mono	10-Bit, Mono	12-Bit, Mono
Dynamic Range	60dB	>50dB	>63dB	>63dB	61dB	>60dB	82dB
Responsivity	16,5/lux-sec	7,0V/lux-sec	8,0V/lux-sec		17V/lux-sec	7,7V/lux-sec	6,5V/lux-s
Output Frame Rate	60	48	60	60	250	43	45
Shutter type	Rolling	Global	Global	Rolling	Global	Global	Rolling
Supply Voltage	1,8/2,8V	1,8/3,3V	1,8/3,3V	1,8/3,0V	2,5/3,3V	1,8/3,3V	1,8/2,8V
Power Consumption	236mW	80mW	200mW	287mW	200mW	375mW	270mW
Op. Temperature	-30°C ... + 70°	-30°C ... + 65°	-30°C ... + 65°	-10°C ... + 60°	-40°C ... + 70°	-40°C ... + 85°	-30°C ... + 70°
Package	BGA63	µCLCC48	CLCC48	152LGA	LCC48	LCC48	ILCC48
Availability	New, since 2015	Since 2010	Since 2013	Since 2009	Since 2006	Since 2013	Since 2014
Price/MOQ	10€/10	125€/1600	125€/1600		160€/1	120€/1	35€/10

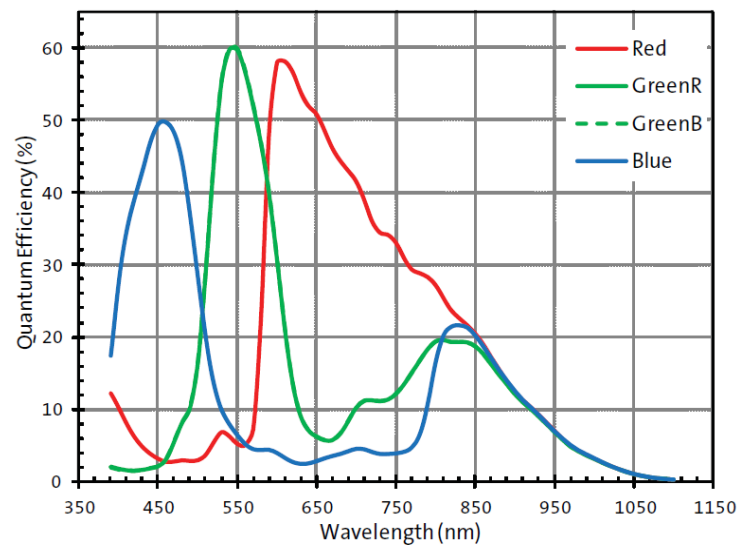
Quanteneffizienz des EV76C454 Bildsensors [E2V20]



Quanteneffizienz des AR130 Bildsensors [ONS19-1]

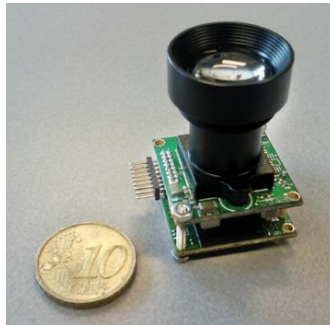


Quanteneffizienz des ASX340CS Bildsensors [ONS19-2]



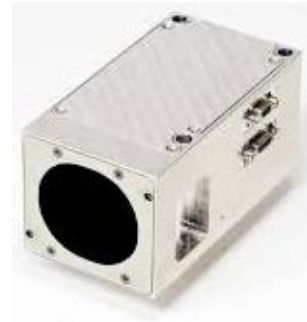
**Anhang H: Auf dem Markt verfügbare Sternsensoren für 1U-, 2U- und 3U-Satelliten
(erstellt im März 2019)**

ST-200, BST / Hyperion



Quelle: www.berlin-space-tech.com

Nano Star Tracker, BCT



Quelle: www.bluecanyontech.com

Parameter	Wert	Wert
Genauigkeit: - pitch, yaw - roll	30 arcsec, 3 σ 200 arcsec, 3 σ	6 arcsec, , 1 σ 40 arcsec, 1 σ
Brennweite oder FOV	16 mm, F1,2	9°x12°
Grenzmagnitude	6	
Sun excl. Angle	35° oder 45°	45°
Ausgaberate	Max. 5 Hz	Max. 5 Hz.
Winkelgeschwindigkeit	Bis zu 1°/s	
Interface	UART oder RS485	RS-422(I2C, SPI)
Steckverbinder	Stiftleiste	
Masse	42 g	350 g
Abmessungen	29 x 29 x 38,1 mm, o. Baffle	50 x 55 x 100 mm
Spannungsversorgung	3,3-5,0V	5,0V±0,1V
Verbrauch	0,650W bei 5 Hz, Peak 1W	< 1,5W
Op. Temperatur	-20...+40°C	
Flugerprob, Preis	>10, 35.000,-€	

AZDK-1, Azmerit



Quelle: www.azmerit.ru

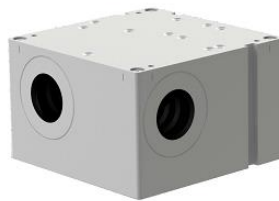
KU Leuven



Quelle: www.cubesatshop.com

Parameter	Wert	Wert
Genauigkeit: - pitch, yaw - roll	10 arcsec, 60 arcsec	2 arcsec, 1 σ 10 arcsec, 1 σ
FOV	22°	9°x12°
Grenzmagnitude	5,5	6
Sun excl. Angle	30°	40°
Ausgaberate	Max. 5 Hz.	Max. 10 Hz.
Winkelgeschwindigkeit	Bis zu 3°/s	
Interface	RS-485	SPI
Steckverbinder	Micro D-Sub 25	
Masse	193g	250g
Abmessungen	56 x 60 x 92,8 mm	45 x 50 x 95 mm
Spannungsversorgung	5V	5,0V
Verbrauch	0,3W, peak 1,25W	<1W peak
Op. Temperatur	-30...+20°C	
Flugerprobt, Preis	4.000€	30.000€

MIST, SPACE MICRO



Quelle: www.spacemicro.com

MAI-SS Space Sextant



Quelle: www.cubesatshop.com

Parameter	Wert	Wert
Genauigkeit: - pitch, yaw - roll	10 arcsec,	4 arcsec 27 arcsec
Brennweite		22mm
Grenzmagnitude		6
Sun excl. Angle	45°	45° mit Baffle
Ausgaberate	Max. 10 Hz.	Max. 4 Hz.
Winkelgeschwindigkeit	Bis zu 5°/s	
Interface	RS-422, CAN, SpaceWire	UART, I2C
Steckverbinder		
Masse	550 g	170 g
Abmessungen	50 x 100 x 100 mm	50 x 50 x 47 mm
Spannungsversorgung		5,0V
Verbrauch	5W	<1,5W
Op. Temperatur		-40°...+80°C
Flugerprobt, Preis		31.000€

ST16, Sinclair



Quelle: www.sinclairinterplanetary.com

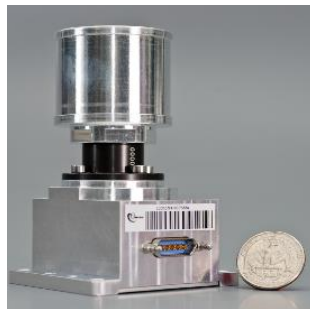
NST-3 Nano



Quelle: www.cubesatshop.com

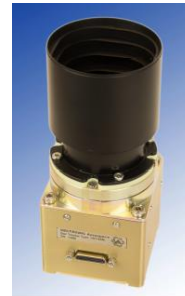
Parameter	Wert	Wert
Genauigkeit: - pitch, yaw - roll	5 arcsec, 3 σ 55 arcsec, 3 σ	5 arcsec, 3 σ 50 arcsec, 3 σ
Brennweite, FOV	16 mm, 15°x20°	15°x12°
Grenzmagnitude		
Sun excl. Angle	34° oder 22°	35°
Ausgaberate	2 Hz	10 Hz
Winkelgeschwindigkeit		< 2,5°/s
Interface	2xRS485	RS422, CAN
Steckverbinder	Micro-D	
Masse	158 g	95 g
Abmessungen	62 x 56 x 38 mm	33 x 33 x 94 mm
Spannungsversorgung	9V...34V	5V
Verbrauch	0,5W, peak 1W	0,6W
Op. Temperatur	-40°C...+50°C	-40°C...+40°C
Flugerprobt, Preis	24, ab 100.000€	30.000€

Auriga, Sodern



Quelle: www.sodern.com

VST-68M



Quelle: www.vectronic-aerospace.com

Parameter	Wert	Wert
Genauigkeit: - pitch, yaw - roll	6 arcsec 40 arcsec	5 arcsec 30 arcsec
Brennweite, FOV		14°x14°
Grenzmagnitude		
Sun excl. Angle	34°	
Ausgaberate	5 Hz	5 Hz
Winkelgeschwindigkeit	<0,4°	
Interface	SpaceWire	2x RS422, 2x CAN
Steckverbinder	Micro-D	Micro-D
Masse	210 g	470 g
Abmessungen	56 x 66 x 94 mm	60 x 60 x 138 mm
Spannungsversorgung	5V	9...40V
Verbrauch	1W	3W
Op. Temperatur	-27°C...+40°C	-20°C...+65°C
Flugerprobt, Preis		

Raumfahrttechnik und Extraterrestrik

Advances in Space Technology and Exploration

In der Schriftenreihe „Raumfahrttechnik und Extraterrestrik“ erscheinen Beiträge zu neuen Entwicklungen aus dem Bereich der Raumfahrttechnik sowie der Erforschung und Nutzung des Weltraums. Diese können sehr gute studentische Arbeiten, Dissertationen, Aufsätze zu relevanten Themen oder Ergebnisse von Projektarbeiten beinhalten.

Herausgeber:
Prof. Dr.-Ing. Hakan Kayal

© 2022

Julius-Maximilians-Universität Würzburg
Prof. Dr.-Ing. Hakan Kayal
Professur für Raumfahrttechnik
Informatik VIII
Universität Würzburg
Emil-Fischer-Str. 32
D-97074 Würzburg
Tel.: +49-931-31-86649
Fax: +49-931-31-81368
hakan.kayal@uni-wuerzburg.de
Alle Rechte vorbehalten.
Würzburg 2022.

Dieses Dokument wird bereitgestellt durch
OPUS Würzburg, ein Publikationsservice der
Universitätsbibliothek Würzburg.

Universitätsbibliothek Würzburg
Am Hubland
D-97074 Würzburg
Tel.: +49-931-31-85906
opus@bibliothek.uni-wuerzburg.de
<https://opus.bibliothek.uni-wuerzburg.de>
Titelblattgestaltung: UB Würzburg

ISSN: 2747-9374

Zitiervorschlag:

Balagurin, Oleksii: Designoptimierung von Sternensensoren für Pico- und Nanosatelliten, Raumfahrttechnik und Extraterrestrik, Nr. 1 (2022).
DOI: 10.25972/OPUS-25896