DEVELOPMENT OF UV ASTRONOMICAL INSTRUMENTS FOR BALLOON AND SPACE PAYLOADS

A Thesis

Submitted for the Degree of

Doctor of Philosophy (Technology)

Submitted by

SARPOTDAR MAYURESH NANDKUMAR

Department of Applied Optics & Photonics
University College of Technology
University of Calcutta

November 2018

To my Guide,

Family, Friends and Teachers...

List of Publications

1. ## Refereed Journal Articles

   (a) **A software package for evaluating the performance of a star sensor operation** - ***Mayuresh Sarpotdar***, *Joice Mathew, A. G. Sreejith, K. Nirmal, S. Ambily, Ajin Prakash, Margarita Safonova, Jayant Murthy*, Experimental Astronomy, Vol 43, Issue 1, pp 99–117, February 2017.

   (b) **Development of Data Acquisition Methods for an FPGA-Based Photon Counting Detector** - *S. Ambily, ,* ***Mayuresh Sarpotdar***, *Joice Mathew, A. G. Sreejith, K. Nirmal, Ajin Prakash, Margarita Safonova, Jayant Murthy* , Journal of Astronomical Instrumentation,Vol 6, 1750002, 2017.

   (c) **A Raspberry Pi-Based Attitude Sensor** - *A. G. Sreejith, Joice Mathew,* ***Mayuresh Sarpotdar***, *Rekhesh Mohan, Akshata Nayak, Margarita Safonova, Jayant Murthy*, Journal of Astronomical Instrumentation,vol. 03, issue. 02, 1440006, 2014.

   (d) **Pointing System for the Balloon-Borne Astronomical Payloads** - *K. Nirmal, A. G. Sreejith, Joice Mathew,* ***Mayuresh Sarpotdar***, *, Ambily Suresh, Ajin Prakash, Margarita Safonova, Jayant Murthy* , Journal of Astronomical Telescopes, Instruments and Systems, Vol. 02, issue. 04, 047001, 2016.

2. ## Refereed Conference Proceedings

   (a) **A generic FPGA-based detector readout and real-time image processing board** - ***Mayuresh Sarpotdar***, *Joice Mathew, Margarita Safonova, Jayant Murthy*, Proc. SPIE 9915, High Energy, Optical, and Infrared Detectors for Astronomy VII, SPIE Astronomical Telescopes + Instrumentation, Edinburgh, UK, 2016.

(b) **Balloon UV experiments for astronomical and atmospheric observations** - *A. G. Sreejith, Joice Mathew,* ***Mayuresh Sarpotdar****, Nirmal K., Ambily S., Ajin Prakash, Margarita Safonova and Jayant Murthy*, Proc. SPIE 9908, Ground-based and Airborne Instrumentation for Astronomy VI, 99084E August 9, 2016.

(c) **An ultraviolet imager to study bright UV sources** - *Joice Mathew, Ajin Prakash,* ***Mayuresh Sarpotdar****, A. G. Sreejith, Margarita Safonova and Jayant Murthy*, Proc. SPIE 9905, Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray, 990533 July 13, 2016.

3. Journal articles not part of this thesis

(a) **Near UV imager with an MCP-based photon counting detector** - *S. Ambily, Joice Mathew,* ***Mayuresh Sarpotdar****, A. G. Sreejith, K. Nirmal, Ajin Prakash, Margarita Safonova and Jayant Murthy*, Proc. SPIE 9905, Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray, 990530 July 11, 2016.

(b) **Prospect for UV observations from the Moon. II. Instrumental Design of an Ultraviolet Imager LUCI** - *Joice Mathew, Ajin Prakash,* ***Mayuresh Sarpotdar****, A. G. Sreejith, K. Nirmal, S. Ambily, Margarita Safonova, Jayant Murthy and Noah Brosch*, Astrophysics and Space Science, Vol 362, Issue 2, pp 11, February 2017.

(c) **Overview of High-Altitude Balloon Experiments at the Indian Institute of Astrophysics**- *Margarita Safonova, Akshata Nayak, A. G. Sreejith, Joice Mathew,* ***Mayuresh Sarpotdar****, S. Ambily, K. Nirmal, Sameer Talnikar, Shripathy Hadigal, Ajin Prakash & Jayant Murthy*, Astronomical and Astrophysical Transactions (AApTr), Vol. 29(3): 397-426 2016.

Presentations

1. **Design, development and functional testing of a star sensor-** ***Mayuresh Sarpotdar****, Joice Mathew, Ajin Prakash, Margarita Safonova, Jayant Murthy*, 3rd COSPAR Symposium, Small satellites for space research, 2017, **Poster presentation**.

2. **High altitude ballooning program at IIA: problems, achievements and experience-** ***Mayuresh Sarpotdar****, Margarita Safonova, Jayant Murthy*, 3rd COSPAR Symposium, Small satellites for space research, 2017, **Oral presentation**.

3. **Development of near and far space payloads-** ***Mayuresh Sarpotdar****, Joice Mathew, S. Ambily, Nirmal K.J., A. G. Sreejith, Margarita Safonova, Jayant Murthy*, India International Science Festival, 2017, **Poster presentation**.

4. **Star Sensor Design: Interface Circuit and VHDL Implementation on FPGA-** ***Mayuresh Sarpotdar****, Neha K, Nikhil PVS, V. K. Agrawal, Jayant Murthy*, Sensors 2016, RCI Hyderabad, **Oral presentation**.

5. **Design and Development of a Star Sensor-** ***Mayuresh Sarpotdar****, Joice Mathew, Ajin Prakash, Margarita Safonova, Jayant Murthy*, 34th ASI Meeting, ASI Meeting, 2016, University of Kashmir, Srinagar, **Poster presentation**.

Abstract

At Indian Institute of Astrophysics, we conduct high altitude balloon experiments for atmospheric and astronomical observations in UV wavelength range. To carry out observations of astronomical sources, a fine pointing system is usually required. One of the prime requirements for such a pointing system is a precision attitude determination sensor. As part of this thesis, I have developed a sensor which determines it's orientation with respect to a fixed inertial coordinate system using bright stars which are fixed in the sky and act as reference points, and hence called a star sensor. We estimated the requirements for the imaging system of the sensor and simulated the sky coverage of the sensor. We also made a MATLAB tool to evaluate the performance of various algorithms required to obtain orientation information from images. Such a sensor has wide application on satellites carrying telescopes, earth observation cameras and high gain directional antennae. The system was designed keeping in mind its applications on a balloon platform as well as it's portability to a satellite platform. Calibration and testing of such a high accuracy sensor are critical to use it efficiently as a part of a pointing control system. In this thesis, we describe the development, calibration and validation of the star sensor from block diagram to fabrication and the methods we used for its calibration. The electronics subsystem designed for the star sensor could be used to readout other image sensors and carry out various low level image processing tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the current times of fast evolution in electronics and manufacturing technology, development of new and complex instruments to observe scientific phenomena has become much faster. This has facilitated observation of astronomical sources from ground based and space based telescopes. Ground based observations of astronomical sources at wavelengths below 3000Å are limited by the ozone layer as it absorbs most of the UV radiations in the earth's atmosphere. A large number of near space and orbital experiments are also designed and fabricated at a fast pace. It is known that large orbital observatories are instrumental in providing accurate observations and statistical studies of remote sensing data as well as astronomical sources. In spite of this, small and lightweight payloads on board high altitude balloons and sounding rockets are attractive due to their short implementation times and low development costs (Sreejith et al., 2015).

The transmission of the atmosphere varies with altitude because of telluric lines of gases present in the atmosphere and down-welling radiance. A model describing the transmittance of different wavelengths of the atmosphere at different heights are known. The transmittance curves at different altitudes viz. Mt. Mauna Kea ($\sim$ 4km), SOFIA a high altitude aircraft observatory ($\sim$ 13km) and high altitude balloon float height ($\sim$ 38km) are depicted in Fig. 1.1 and are described in detail in Hibbits et al. (2013). A dramatic im-

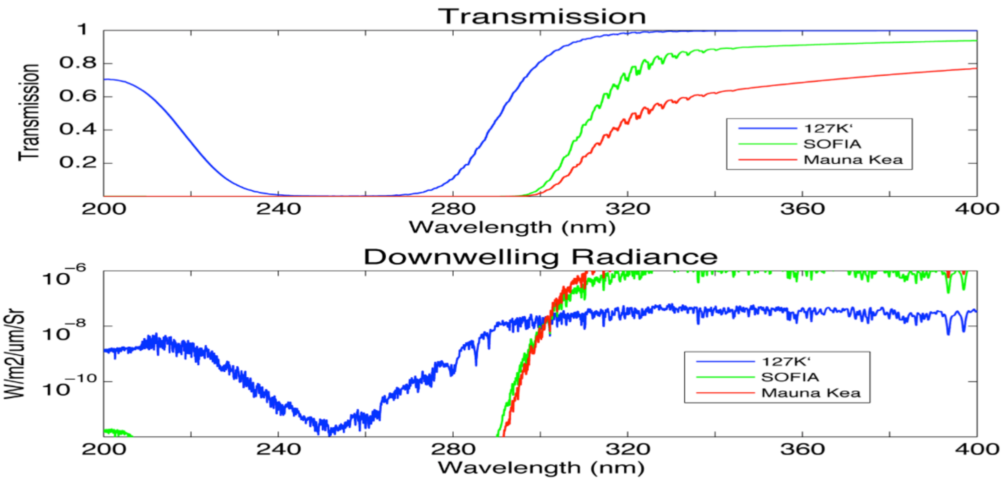


Figure 1.1: Near UV vertical transmission and daytime downwelling radiance from *Hibbits et al., 2013* (Image credit: *Hibbits et al., 2013*).

provement in atmospheric transmittance occurs in the wavelength band of 280-400 nm at higher altitudes. In-spite of this increase in transmittance, high altitude balloon platforms do not enable observations between 240-280nm. However, above 35km the atmospheric absorption drops drastically to about 8%, and a window of observation opens up enabling observations above 190nm-240nm(Navach et al., 1973). High altitude balloons have been used for a number of scientific observations of astronomical sources in high energy and infra-red (IR) wavelength bands, still very little work has been carried out in the near ultraviolet (NUV) window from 280-400 nm. This wavelength band includes various spectroscopic emission lines important in atmospheric chemistry. These emission lines are important to understand the contribution of the atmosphere as foreground in NUV astronomical observations. In addition it helps to study the atmospheric processes and understand the greenhouse gases and climate change.

At float heights in the stratosphere, the absence of 99% atmosphere enables diffraction limited imaging similar to that of space observatories, at a much lower cost. For example, with sufficient pointing stability and accuracy, a UV telescope (200–400 nm) located at these altitudes with an aperture of just 6-inches and a 1K×1K CCD array, can provide wide-field images with

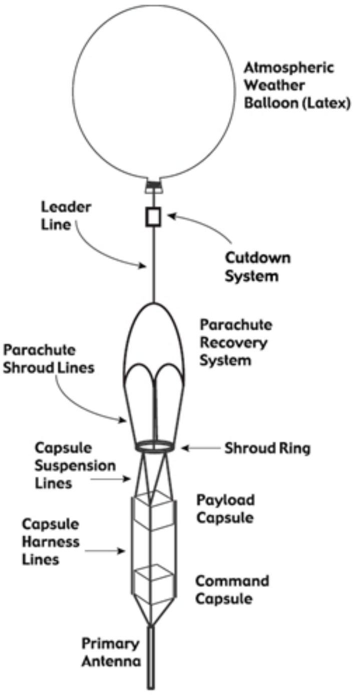


Figure 1.2: Components of a high altitude balloon platform (Image credit: *Sreejith et al. 2016*)

FWHM better than 1″ (Fesen et al., 2015). Additionally, space-based observatories have strict constraints on observability and pointing direction, owing to detector safety reasons and operational requirements. These constraints on Sun/Moon angles limit the observations of stellar sources due to presence of bright solar system objects. This is not the case with balloon based observatories which can be launched for observing a few sources for longer durations in a single launch, mainly because their trajectory is not orbital in nature.

1.1 Balloon observatory

At Indian Institute of Astrophysics, such high altitude balloon experiments are regularly conducted to study diffuse sources (like zodiacal light or airglow) in NUV wavelength range (Margarita Safonova et al., 2016). Various experiments to understand atmospheric absorption and emission features, known as air-glow lines were carried out using payloads with platforms having coarse pointing capability (Sreejith et al., 2016). The setup required for this, constitutes of a space segment and a ground support segment. The space segment includes various components as shown in Fig. 1.2. Major components in this space segment are: balloon, a cut down system/ flight termination unit (FTU), a parachute, the payload capsule and the telemetry/command capsule. A hydrogen filled balloon is used to generate adequate lift to raise the payload weight up-to the desired altitude in the atmosphere. The FTU works as a fail-safe mechanism to ensure successful retrieval of the payload. This is essential in situations where the balloon does not burst after reaching the maximum height, to avoid it from drifting far away with winds. The parachute enables controlled descent of the payload thus avoiding collision damage. The payload capsule is made using Styrofoam which is a thermal insulator and provides necessary thermal insulation to the instruments placed inside and maintains a suitable operating temperature for the instruments. The command capsule carries a radio tracking system to relay the balloon GPS coordinates and altitude to the ground station in the form of a beacon signal. This enables tracking of the balloon train and successful payload recovery. The ground support segment consists of an omni-directional antenna and a radio receiver setup. This setup decodes the transmission from the balloon beacon sent by the telemetry/command capsule.

Observations of the air-glow in the atmosphere are essentially random directional and do not require the optics to be pointed in a specific direction. However, subsequent astronomical observations require an accurate pointing system which directs the telescope to a certain stellar object in the sky with minimal jitter and better accuracy. Such a pointing system along with the NUV telescope would constitute a functional balloon borne NUV astronomical observatory. An illustration of the complete pointing system is as shown in the Fig. 1.3. It consists of an NUV telescope (blue), star sensor (orange), spectrograph(gray) and all of these instruments mounted on a 2-axis pointing system used to orient them in the required direction. We are developing such a balloon borne NUV astronomical observatory. This requires a highly accurate 2 axis pointing system. Nirmal et al. (2016) demonstrates the working

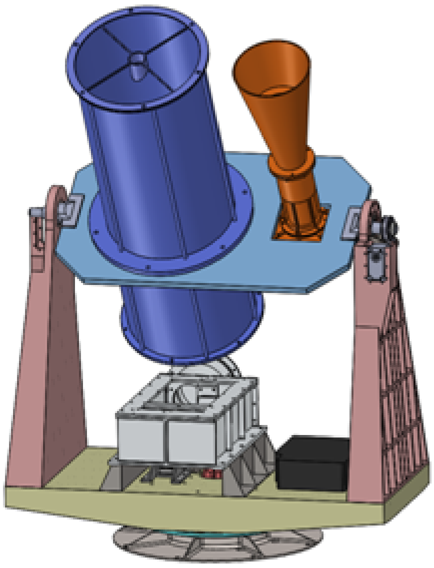


Figure 1.3: Balloon borne observatory platform. This platform essentially consists of an NUV telescope, a spectrograph at the backend of the telescope, a 2-axis pointing system, and a star sensor for fine pointing knowledge. (Image credit: *Sreejith et al. 2016*)

of a 2 axis pointing system with an accuracy of about 0.5° which is sufficient for atmospheric observations for air-glow in conditions where it is necessary to point the telescope to an extended source, but inadequate for astronomical sources. Hence a more accurate pointing system is desired which would perform at least an order of magnitude better in accuracy. One of the main limiting factors for the accuracy of the pointing system is the accuracy with which the attitude sensor estimates the attitude. Thus, it is desirable to improve the accuracy of the attitude sensors by at least an order of magnitude. Astronomical observations being carried out at night, the bright stars in the sky are always visible. These bright stars are stationary at their place in an inertial coordinate system and hence, provide standard reference points. The stars can be identified based on the patterns they form and a final orientation estimate can be obtained using these known patterns. A sensor which determines orientation by using these star patterns as reference points is called as a star sensor. This thesis mainly focuses on the design, development, calibration and rigorous functional testing of such a star sensor.

1.2 Thesis outline

Chapter Two: Background and simulations

This chapter describes the selected catalogs for the star sensor. A sky simulation is done by selecting a bright star catalog and using it to obtain distribution of the stars in the sky This in turn was used to estimate sky coverage of cameras with different field of views and limiting magnitudes. Finally a methodology to select physical parameters of the imaging optics in the star camera is described. The effect of using various criteria to estimate sky coverage are also described in this chapter.

Chapter Three: Algorithms and software tool to evaluate performance

Various coordinate systems which are important to be understand the functioning of the star sensor are described in the beginning of this chapter. The star sensor involves various image processing tasks to be undertaken in real time on the images being captured by the star camera. Various simulations were done to validate the required operation of the star sensor algorithms. These simulations are described in detail in this chapter. Various parameters like sky coverage, memory requirement for the algorithm implementation and timing of different steps of the algorithms were evaluated in the process thus giving inputs for the hardware design of the star sensor.

Chapter Four: Design and analysis of *StarSense*

After studying the catalogs to be used in the star sensor and analyzing the software to be used on the star camera, an understanding about the requirements of the imaging system of the star sensor were established. Different blocks of the hardware (viz. optics, electronics and structure) were designed and analyzed to meet these requirements. The design process of these hardware blocks of the star sensor is discussed in this chapter.

Chapter Five: Calibration of the *StarSense*

After having assembled the star camera, determining its camera parameters is crucial for precise operation of the star sensor. Also understanding the noise in the image sensor is crucial for optimizing the exposure for various limiting magnitude stars. Various tests were implemented to determine these camera parameters and the CMOS image sensor calibration and thus calibrate the star camera. These tests were carried out in laboratory as well as using low cost, easily available apparatus to validate the low cost testing procedures used in the field environments. The results were validated using standard calibration procedures too. This enabled identifying ways to calibrate the camera accurately without the standard laboratory apparatus. These tests and their results are discussed in this chapter.

Chapter Six: Functional validation of the *StarSense*

After calibrating the camera and identifying its intrinsic parameters using both laboratory apparatus as well low cost apparatus, we implemented similar tests for the complete functioning of the star camera as a star sensor. We carried out the functional validation tests in standard lab environments using a star simulator as well as by co-aligning the star camera with a tracking telescope. The controller of the tracking telescope could accurately determine the pointing direction of the camera thus giving a reference to validate the output of the star sensor with. We describe the functional validation tests in this chapter.

Chapter Seven: Conclusion and future work

The star sensor consisting of a very sensitive star camera, can also be used for photometry of some bright variable stars and provide valuable scientific information. We evaluated its suitability for our balloon platform and determined its utility as a scientific camera and concluded its use cases other than a star sensor, which are described in this chapter. After having fabricated, tested and validated a star sensor to adapt to a pointing system for a balloon borne telescope, making it suitable for small satellites is the next step. Considerations to make this star sensor suitable for space are discussed in this chapter and a future pathway leading to development of a star sensor for nanosatellites is projected.

Chapter 2

Background and simulations

2.1 Introduction

As discussed in the previous chapter, astronomical observations on a balloon observatory require a pointing system which can enable a telescope to be pointed at a desired astronomical source to continuously stare at it against vibrations and disturbances. General purpose IMU sensors and servo motor actuators can be used in such a pointing system to achieve an accuracy of $\sim 0.5°$ (Nirmal et al., 2016). To achieve pointing accuracies better than $\sim 0.5°$ a more accurate orientation sensor as well as a more accurate actuator is required. As a first step of the development of an improvised pointing system we chose to develop a fine and absolute attitude sensor. This pointing system is required to orient a telescope to stars which are fixed at their positions in the sky with respect to an equatorial coordinate system. It is best to determine the pointing direction of the telescope with respect to these fixed stars itself. Such a sensor which determines it's orientation with respect to the stars is called a star sensor. We named the sensor we have developed as *StarSense* and we describe the development process, testing and functional validation of *StarSense* in this thesis. A similar sensor can be used in satellite and space telescopes where the sensor determines the orientation of the telescope with respect to the fixed stars in the sky. The pointing system in a space observatory would include reaction wheels, control moment gyroscopes or similar other momentum transfer devices as actuators to control the orientation of the satellite (Ali Siahpush et al., 1988).

2.2 Attitude control system & star sensor

A control system which helps the telescope payload on a balloon platform or a satellite to orient in the desired direction against surrounding and internal disturbances is called an attitude control system. It is indispensable for satellites requiring pointing capabilities. A typical attitude control system consists of attitude sensor, a controller and actuators. Various attitude sensors are used in different operational conditions for example in space, a combination of sun sensor, magnetometer and inertial measurement unit is used for coarse attitude sensing and a star sensor is used for fine and absolute attitude sensing. For ground based pointing systems, encoders, magnetometers and inertial measurement units are used. The data from these sensors is fused together to get the attitude/orientation solution. Typical accuracies that can be achieved using various sensors are as tabulated in Table 2.1. Various actuators are used in different operational cases of the attitude control system viz. servo motors and stepper motors in case of balloon observatories, magnetic torquers, reaction wheels, and control moment gyroscopes etc. in case of satellites. The magnetic torquers interact with earth's magnetic field and generate a torque to orient the satellite in the desired direction. Reaction wheels and control moment gyroscopes use a rotating mass and a mechanism to utilize the inertia of the rotating mass to convert to a torque in a particular direction whenever disturbances are generated. For ground based and balloon based observations a 2 axis pointing system consisting of actuators like servo motors and direct drive motors which are used to actuate the azimuth and altitude directions of the telescope.

The telescopes used for astronomical observations have a field of view of

Table 2.1: Attitude control accuracies that can be achieved using different sensors

Sensor	Typical Accuracies (deg)
Magnetometer with coarse sun sensor	5
Magnetometer with fine Sun sensor	0.2
Horizon sensor	0.3
Star tracker	0.01

the order of a few tens of arc minutes. Hence, the pointing system required by these telescopes to work correctly should be more accurate. To achieve such a pointing system the first step is to build a sensor which can sense the orientation of the platform with accuracy better than the target pointing system accuracy. An attitude sensor built using a generic IMU could provide attitude measurement accuracies of the order of $\sim 0.5^\circ$. Hence to be one magnitude of accuracy better than these sensors, we considered a design accuracy requirement for *StarSense* of the order of $\sim 30''$. Also telescopes are to be used to observe stars which are fixed in the earth centered inertial (ECI) coordinate system. So a sensor which determines orientation with respect to the ECI coordinate system is desirable. *StarSense* outputs the orientation information with respect to the ECI coordinate system.

In case of satellites, especially in the modern era of nano-satellites, such attitude control systems are desirable to be light-weight, inexpensive, small volume and low-power consuming. Similarly in case of a balloon borne payload platform, it is desirable to have a light-weight, low cost attitude control system. Figure 2.1 shows a basic conceptual schematic of a pointing system which would enable astronomical observations from a balloon borne payload platform. We are in the process of building an attitude control system for a balloon borne payload platform (Fig. 1.3) to enable astronomical observations using an ultraviolet telescope during flight.

The problem statement of an attitude sensor is essentially to determine the relative rotation between two coordinate systems. The sensor has to determine the relative rotation between the sensor coordinate system and a fixed reference inertial coordinate system (Fig. 2.2). The coordinate system defined by X, Y, Z can be assumed to be the reference inertial coordinate system where as

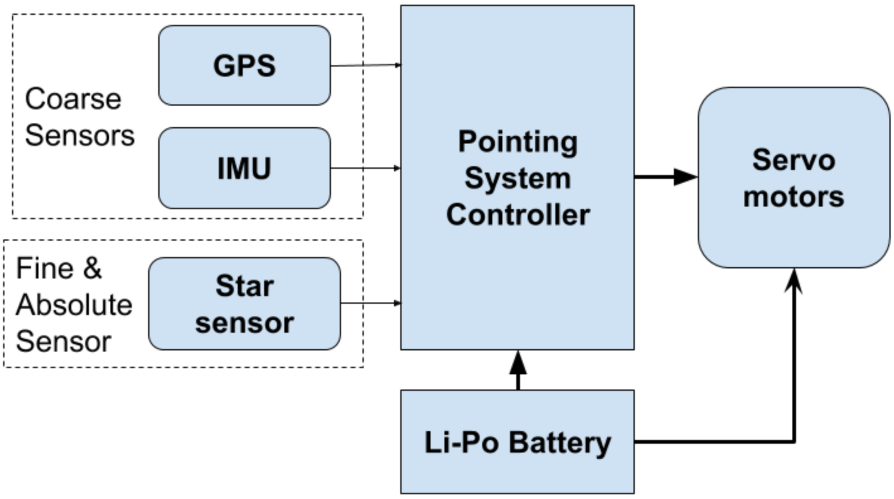


Figure 2.1: Systemic block diagram of an attitude control system for the balloon observatory platform

the coordinate system defined by G_x, G_y, G_z can be assumed to be the sensor coordinate system. Finding the rotation matrix between these coordinate systems is to basically determine the marked angles in the figure. This rotation information can be represented in terms of rotation matrix, Euler angles, or quaternions.

2.3 Star Catalog

The locations of the stars are fixed in the sky considering an ECI coordinate system. The typical drift rates of stars defined by their proper motion for some of the fastest moving stars are of the order of $\sim 1''/year$. This motion is negligible and cannot be detected in wide field instruments like *StarSense* . In addition to that, the time duration of the balloon launch is much smaller than a year resulting in miniscule movement of stars in the sky with respect to the ECI coordinate system. Hence, the stars are considered essentially stationary with respect to the ECI coordinate system.

A study of the catalog of stars in ECI coordinate system is the ground work for proper operation of *StarSense* . The stars can be used as the reference points to determine orientation. To enable this, the position and bright-

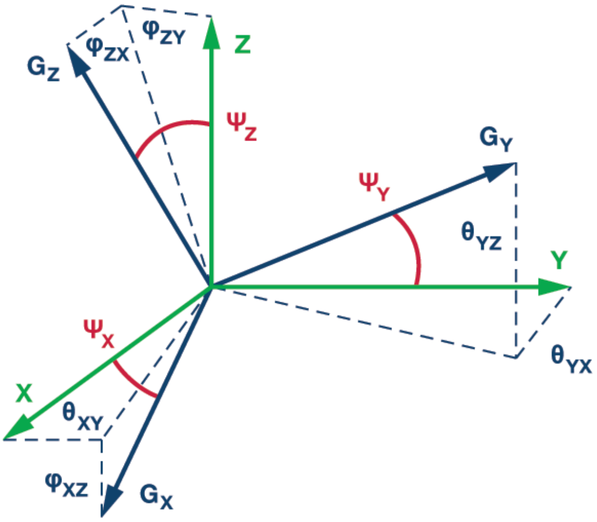


Figure 2.2: Finding the rotation between a reference and sensor coordinate systems is the main concept of operation of an attitude sensor. The green coordinate system XYZ shows a standard reference coordinate system where as the blue coordinate system $G_xG_yG_z$ shows the rotated sensor coordinate system

ness of these stars/ reference points should be known to the sensor apriori in a structured format which is called a catalog. Many different star surveys have been conducted which have determined various parameters viz. apparent magnitude, absolute magnitude, coordinates in a fixed epoch, proper motions, color indices etc. for these reference stars and the results have been published in catalogs available in open domain. We have considered Hipparcos Bright star catalog for this ground work. The sensor imaging these bright stars would be limited in integration time depending on the data rate required from the sensor. The size of the optics used would limit the amount of photons collected from a particular star during a given integration time. Thus there would be a limiting value of visual magnitude for the sensor, to be determined using photometric analysis. We did various simulations using the star catalog to determine the required limiting magnitude which would enable proper operation of the star sensor.

2.3.1 Properties of a star catalog

The stars in the catalog demonstrate certain statistical properties while considering various filter criteria. The number of stars brighter than a certain limiting magnitude increases drastically as the limiting magnitude increases. This results in a large table of stars to be stored on-board for real time processing. Thus, the on-board memory required to store the star data increases. Along with the increasing catalog size, the time required for identification of the stars increases drastically as the number of stars in the catalog increases. This is because each pattern needs to be compared with the entire catalog to be identified uniquely, thus requiring more numbers of comparisons. A statistical analysis shows that the variation of number of stars with the limiting magnitude can be expressed by the equation 2.1

$$N = 6.57 \times \exp^{1.08M_v} \tag{2.1}$$

where N is the number of stars and M_v is the limiting visual magnitude selected for the imaging optics.

The actual number of stars brighter than a given limiting magnitude after processing from the Hipparcos catalog are as shown in the Fig. 2.3. Clearly the number of stars and correspondingly the entries in catalog increase exponentially, we can limit our catalog search to a maximum of about 6.5 magnitude. This would provide enough number of reference guide stars in a limited star identification time.

After knowing the number of stars in the sky brighter than a certain limiting magnitude, their distribution in the sky is to be evaluated. The limiting magnitudes we are considering would enable only stars from our galaxy to be tabulated in the catalog. We know that the milky way galaxy is spiral in structure and thus, the distribution of the stars in the sky would suggest the galaxy structure. Fig. 2.4 shows a distribution of the stars in the sky at different limiting magnitudes in an aitoff projection.

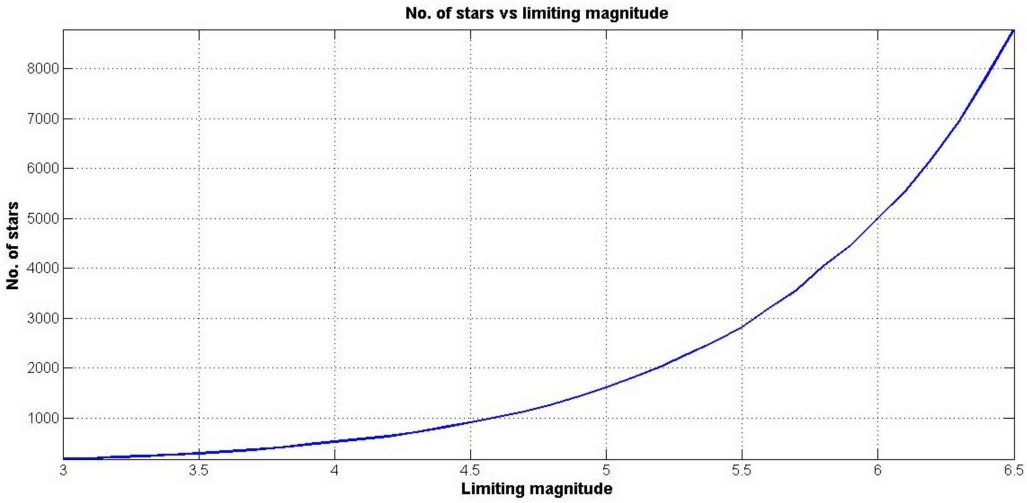


Figure 2.3: Variation of star population with increasing limiting magnitude obtained after filtering a Hipparcos star catalog

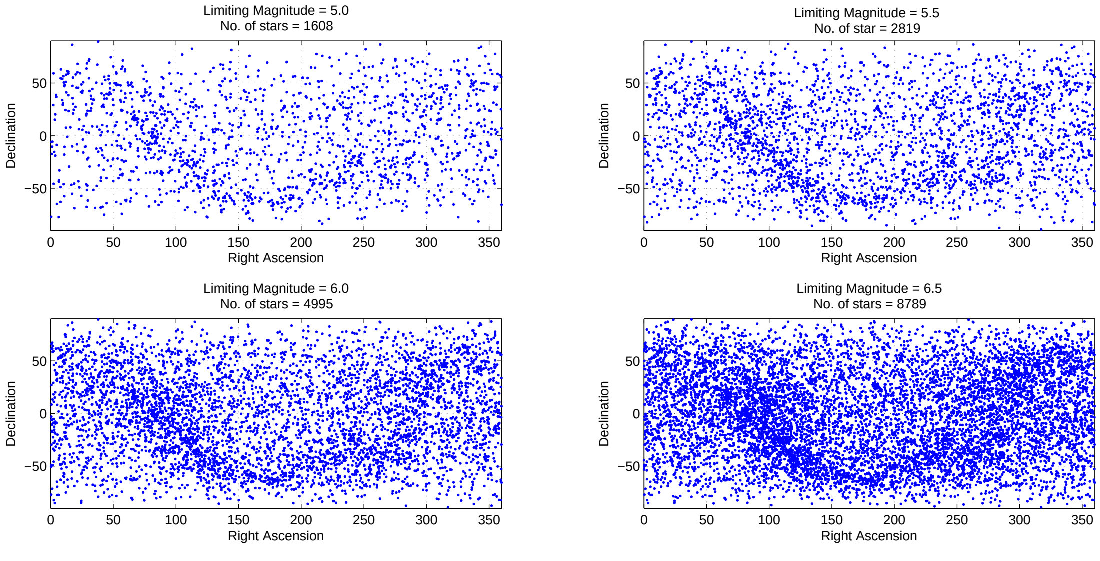


Figure 2.4: Visualisation of star distribution in the sky in aitoff projection at different limiting magnitudes obtained after filtering Hipparcos star catalog

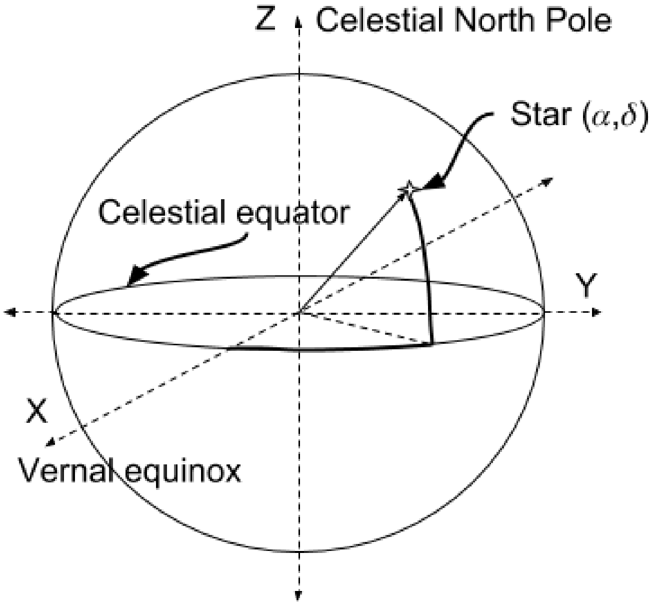


Figure 2.5: Visual definition of earth centered inertial (ECI) coordinate system

2.3.2 Unit vectors in ECI coordinate system

The sky is modeled in the form of a unit sphere (i.e. a sphere with a radius of 1 unit) for the sake of the catalog, where the distance of the stars is not considered but their positions in terms of angles from a fixed coordinate system are considered. In fig. 2.5, the XYZ coordinate system depicts this coordinate system which is called earth centered inertial coordinate system. The reference directions considered in this coordinate system are the vernal equinox which corresponds to X axis, the axis of rotation of earth which is the north celestial pole which corresponds to the Z axis and the Y axis completes the triad. The position of any star can be broken down to 2 component angles in this coordinate system: right ascension and declination. The declination is the angle of the star from the celestial equator and the right ascension of the star is the angle between the vernal equinox/X axis and the footpoint of the star on the celestial equator.

The position of the star given in terms of right ascension and declination has to be converted into unit vectors to determine the direction of the stars in the 3-dimensional coordinate system. Eq. 2.2 can be used to convert the position coordinates of the star from right ascension and declination form to unit vector form. These unit vectors are also useful to calculate angles between any two stars by using the dot product of the unit vectors of the pair of stars.

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \cos\delta . \cos\alpha \\ \cos\delta \sin\alpha \\ \sin\delta \end{bmatrix} \tag{2.2}$$

2.3.3 Partitioning of a star catalog

As seen from the previous section, the distribution of the stars in the sky is not even. Hence, a way of partitioning the sky in smaller segments is required which would represent model of images captured by the star sensor. It is also required to scan the complete sky in the process of partitioning for sky simulations. Various methods can be used to partition the sky into smaller segments based on geometrical properties, imaging system properties etc. Some of the current methods available for such partitioning are as listed below (Guangjun Zhang, 2017):

- Declination zone method
- Cone method
- Sphere rectangle method
- Inscribed cube method

Out of these, predominantly the sphere rectangle method was used in various simulations regarding the star catalogs as described further in the thesis.

2.4 Sky coverage simulation

One of the important parameters to determine for evaluating a star sensor operation is to determine the sky coverage of the *StarSense* . This is defined as the area of the sky in which the *StarSense* will be able to identify the stars uniquely and calculate the quaternion correctly from lost in space mode *i.e.* when the sensor does not have any idea of it's current pointing direction.

To estimate this we first partitioned the star catalog by forming overlapping fields in the sky which would enable us to cover the complete sky. We assumed that for a successful identification of stars there should be enough patterns (*i.e.* angles between stars) to be able to match with patterns in a catalog. A set of 5 stars would provide about 10 different patterns to match with patterns in catalog. Hence, we assumed that there should be a minimum of 5 stars in any field to successfully identify the stars in the field. Furthermore the identities of these stars from the images captured are sufficient to calculate the attitude/orientation matrix with respect to the ECI coordinate system.

Considering this 5 star constraint as the limiting constraint, a simulation was carried out to determine the number of fields in the sky which satisfy the 5 star criterion at different combinations of limiting magnitudes and fields of view of the optical system. Thus, there are 2 variables to be determined using this simulation. As the limiting magnitude goes on increasing, there are drastically more number of stars visible but the required optics needs to be more sensitive which translates into larger lenses and a heavier system. As the field of view of the system increases the number of stars in each field go on increasing. This results in an increased sky coverage, albeit at the cost of accuracy. For a wider field of view the plate scale of the instrument is more, thus causing it to become less accurate in determining the attitude. This simulation helps in selecting a proper combination of these 2 variables to obtain a suitable sky coverage. Fig. 2.6 shows the result of the simulation. X axis shows the field of view and y axis shows the sky coverage obtained. Different colored lines in the graph show the sky coverage vs field of view at different limiting magnitudes of the sensor.

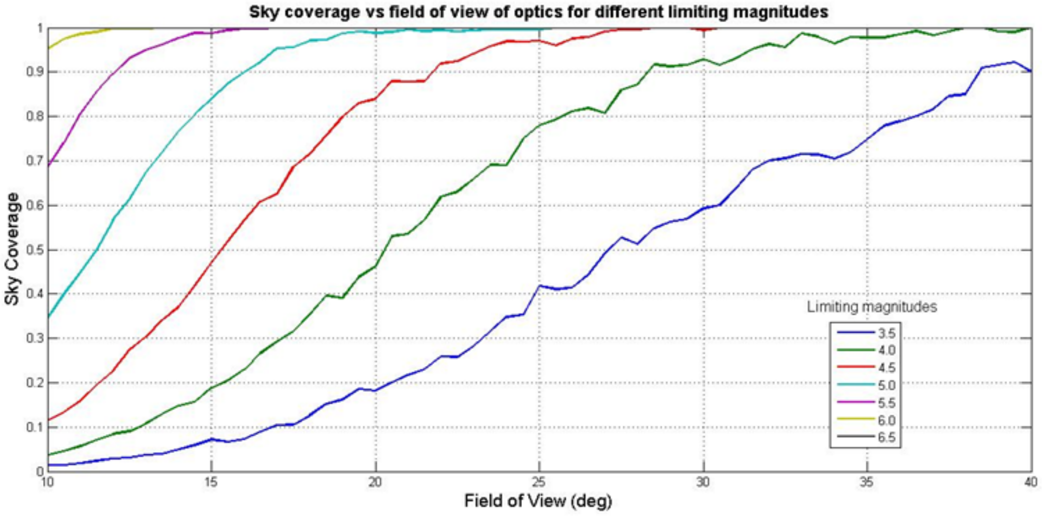


Figure 2.6: Sky coverage at varying field of views and limiting magnitudes. A sky coverage of more than 0.9 is desirable for suitable operation of the star sensor.

2.5 Conclusion

From careful observation of the graph, different operating points for the star sensor can be selected which allow high sky coverage depending on the end design requirements of the star sensor in terms of accuracy and weight constraints. As mentioned previously, our target was to achieve an attitude determination accuracy of about 30″ which would be a good starting point for the design of the attitude control system required for the high altitude balloon. To obtain this pixel scale, the field of view considering a $1k \times 1k$ image sensor would be of the order of $8^\circ \sim 12^\circ$. To achieve a sky coverage of $\sim 90\%$ a limiting magnitude of ~ 6.0 is required as can be seen from the figure. At brighter limiting magnitudes, the field of view required to achieve $\sim 85\%$ sky coverage would be higher i.e. of the order of $\sim 20^\circ$, which will result in larger pixel scale resulting in lowering of attitude accuracy. On the other hand for fainter limiting magnitudes, the lens weight would increase to get a larger aperture area. Thus we obtained the suitable physical parameters required for the *StarSense* optics, from these simulations.

Chapter 3

Algorithms and software tool to evaluate performance

3.1 Introduction

An attitude/orientation control system is necessary for all satellites/ high altitude balloon observatories with varying degrees of pointing accuracy. Satellites with a high-gain directional antenna, a telescope, or an Earth-imaging instrument as payloads require highly accurate (few arc minutes or arc seconds) pointing system. Determining the current pointing position is of primary importance in such applications and it is usually achieved using star sensors: a wide-FOV camera with real-time image processing capability which processes the live image data to obtain an attitude quaternion that describes the rotation of the sensor coordinate system with respect to the Earth-centered inertial (ECI) coordinate system. This is achieved by applying multiple algorithms on the image in sequence (Spratling, B., 2009). We describe here the development of a low-cost star sensor *StarSense* built using off-the-shelf components. This can be deployed on minisatellites, CubeSats and high altitude balloon platforms. We have used the centroiding algorithm for finding centroids of stars seen in the image, the Geometric Voting algorithm for star pattern identification, and the Quaternion Estimator (QUEST) algorithm for quaternion estimation. To determine how each algorithm performs in terms of sky coverage, memory requirements, calculation time, and so on, we simulate the algorithms along with idealized and real hardware parameter inputs using a specially developed test software package.

In this chapter, we discuss a software package intended to evaluate the performance of algorithms to be implemented on *StarSense*. This software package was developed to evaluate any star sensor with different hardware parameters, such as the focal length of the imaging setup, the field of view (FOV) of the camera, angle measurement accuracy, distortion effects, and others. As discussed in the previous chapter, there can be multiple ways in which high sky coverage can be obtained. This software package helps in evaluating additional operational parameters for a particular selection of physical parameters of a star sensor. It is written to run under MATLAB programming language, due to the simplicity of scripting and its excellent capability of visualizing the results. The implementation of these algorithms on actual star sensor electronics hardware will be done in C, keeping in view its easy portability to other platforms.

3.2 Software implementation

All codes in this package are written in MATLAB (Mathworks, 2013) due to its simplicity in scripting and excellent capabilities for visualizing the data. MATLAB (matrix laboratory) is an environment and fourth-generation programming language for numerical computations and simulations developed by MathWorks[1]. Fig. 3.1 depicts the flowchart of the software of a lost in space mode operation of the star sensor. This flowchart shows the sequence of different algorithms that are used to calculate the quaternion from a star field image. A custom FPGA IP core module designed for the Star 1000 image sensor, would capture image from the detector and dump it into the on-board

[1] MATLAB is a licensed product which can be obtained from `https://in.mathworks.com/store`

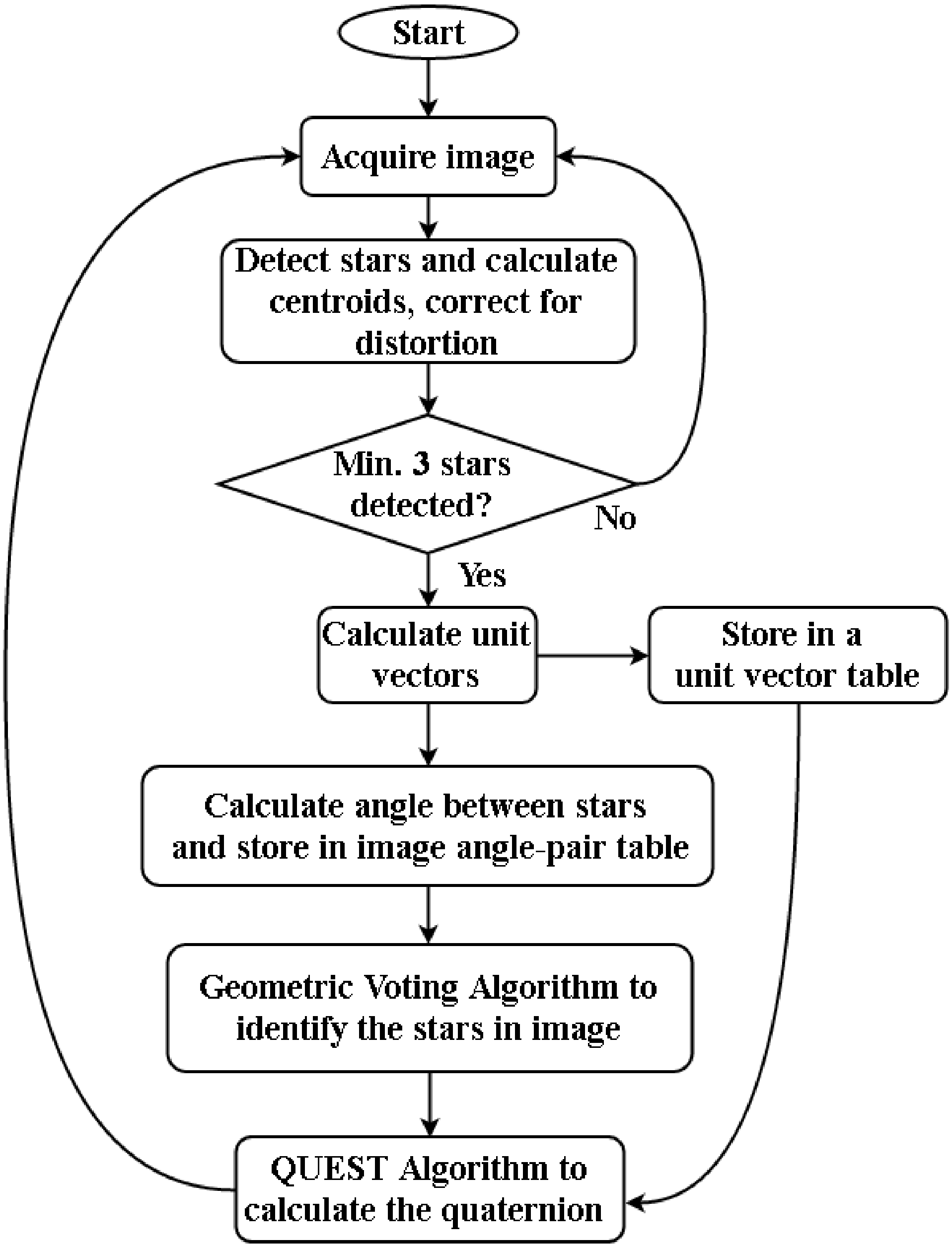

Start
Acquire image
Detect stars and calculate centroids, correct for distortion
Min. 3 stars detected?
No
Yes
Calculate unit vectors
Store in a unit vector table
Calculate angle between stars and store in image angle-pair table
Geometric Voting Algorithm to identify the stars in image
QUEST Algorithm to calculate the quaternion

Figure 3.1: A flowchart of the overall software architecture. This shows a single lost in space mode of operation of the star sensor. The sequence of implementation of different algorithms on the input images are highlighted in the flowchart.

external random access memory (RAM). After image acquisition, a microcontroller, instantiated on the FPGA, starts a process of identification of regions corresponding to stars. Pixels having values higher than a predefined threshold are prospective stars. Stars are differentiated from other objects based on the number of pixels in the region. Centroids, which are essentially the (x, y) coordinates of the centers of the stars in the image plane, are then calculated. These are converted into unit vectors in the sensor body coordinate system using Eq. (3.3) below. The angles between stars are estimated by taking a dot product between every pair of stars detected in the image. Subsequently, the Geometric Voting algorithm identifies the stars in the image by finding a match for the angle pairs from the angle pairs in the catalog. A binary search algorithm (Barjatya A., 2005) is used to search the angles from the catalog table. Lastly, the QUEST algorithm is used to calculate the quaternion of rotation between the ECI coordinate system and the sensor body coordinate system.

After understanding the required algorithm for the complete solution, separate simulations were conducted on the MATLAB implementation of each algorithm to estimate their performance. The simulations were carried out in sequences as shown in their respective flowcharts which are mentioned below:

- Centroiding algorithm – Fig. 3.4
- Geometric voting algorithm – Fig. 3.5
- QUEST algorithm – Fig. 3.8

In the following sections, we describe the algorithms, the simulations used for their performance estimation and the results.

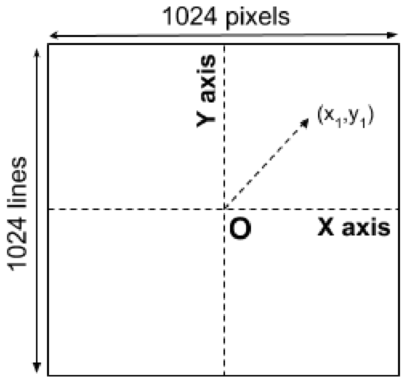


Figure 3.2: Visual definition of image plane coordinate system. The X- & Y- axis of the image sensor serve as the X- & Y- axis of this coordinate system.

3.3 Algorithms

3.3.1 Coordinate systems

The procedure for obtaining attitude quaternions from images needs clear definition of various coordinate systems used. We use the following coordinate systems in the process:

- Image plane coordinate system
- Sensor body coordinate system
- Earth-Centered Inertial coordinate system

The 2-dimensional image plane coordinate system describes location of points in the detector plane. The point of intersection of the optical axis with the image plane (also known as principal point of the optical system) forms the origin of this coordinate system. The x and y axes are in the directions of the columns and rows of the image sensor area, respectively. The principal point of the optics is identified during the calibration of the camera. The units of this coordinate system are pixels. Distances in real world units for this coordinate system can be obtained using the pixel pitch of the sensor. Fig. 3.2 shows an illustration of the image plane coordinate system.

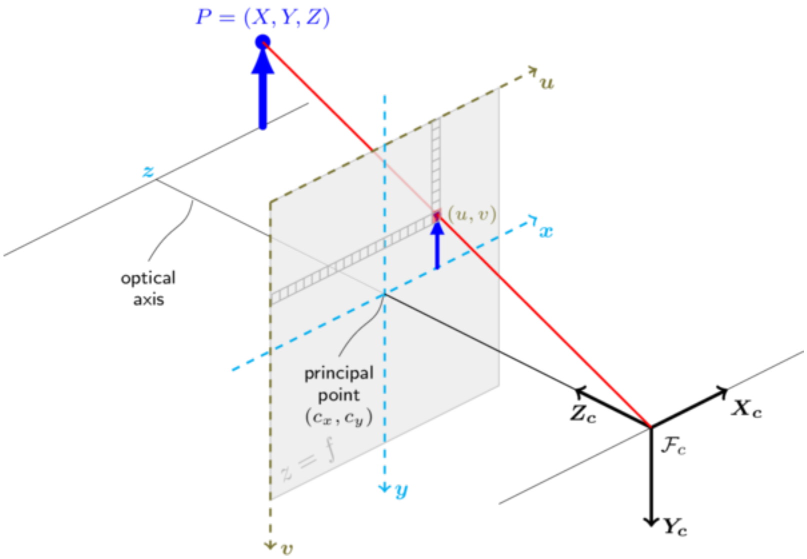


Figure 3.3: Visual definition of sensor body coordinate system. The gray area identifies the location of the image sensor. $X_cY_cZ_c$ mark the coordinate axes of the sensor body coordinate system.

The 3-dimensional sensor body coordinate system has its origin at the optical center. The z-axis is the direction of the lens bore sight (Z_c as shown in Fig. 3.3), and its x and y (X_c and Y_c) axis are aligned with the x and y axis of the image plane coordinate system. The distance up-to the principal point on the image sensor plane from the origin of the sensor body coordinate system is the focal length of the sensor. Sensor body coordinates are expressed in Cartesian form. Unit vectors of the stars in a given image are calculated in this coordinate system and are further used in Geometric Voting algorithm and QUEST algorithm.

The 3-dimensional Earth-Centered Inertial coordinate system is fixed in inertial space and thus used as the reference frame to calculate the quaternions. The z-axis points towards the North Celestial pole, and the x-axis is in the direction of vernal equinox (refer Fig. 2.5). This is a unit-sphere coordinate system, and the position of any star is expressed in terms of only two angles: Right Ascension (RA), and Declination (Dec). The epoch in which the ECI coordinates of stars in the catalog are described, should be clearly mentioned, because the RA/DEC of stars change over time due to precession, or proper motion.

3.3.2 Centroiding algorithm

The image of a star on the detector is slightly defocussed and spread over multiple pixels. This effect is due to the point spread function (PSF) of the lens used which is purposefully designed so that the PSF can be sampled properly. Therefore, the position of a star on the image plane is identified by calculating the centroid of the region where the star is imaged. The spread of a stellar image helps in filtering stars from erroneous hot pixels and extended objects. A region with very few bright pixels is a region of hot pixels or cosmic ray hit, and a region with more number of bright pixels than expected is an extended object. The optical design is made such that a star image is spread over a 4×4 (16) pixel area. Therefore, any bright region which contains less than 10 pixels is rejected for being a hot pixel, and any bright region occupying more than 20 pixels is rejected for being an extended object. This filtering happens during the readout process and, thus, faulty star images do not propagate further in the calculation flow. Another advantage of spreading a star image is that the centroid coordinate can be calculated with sub-pixel accuracy, as opposed to the case when the centroid coordinates are integer numbers of the pixel position in x and y directions.

The centroiding algorithm provides the centroid location in the image plane coordinate system. The computing procedure consists of three processes (Erlank A. O., 2013):

- Image plane search
- Segmentation/region growing
- Centroiding

To identify a region as a star, we must first scan over the complete image area for pixels brighter than a pre-determined threshold value. This process is called the image plane search. The threshold value is identified during the calibration and real sky imaging. Since the size of the whole image is 1K×1K, and each pixel is read from memory in approximately 10 ns, scanning of the complete image (1 million pixels) takes significant portion of the total time, from image acquisition to the quaternion calculation. We check every alternate pixel to reduce the net time while still detecting stars successfully.

On identifying a pixel with a value more than the pre-determined threshold, we start a process called region growing. In this process, we check neighboring pixels for values greater than the threshold. On finding a neighboring pixel brighter than the threshold, we repeat the same region growing procedure for the neighboring pixel. During this looping routine, we also keep a count of the number of pixels in the region. Filtering of hot pixels/cosmic ray hits and extended objects happens at this stage. At the end of the loop, a region in the image which belongs to a star is identified. This region is allotted a number, and the image plane search is continued. At the end of the image plane search, all regions corresponding to stars are identified and given an image ID number.

After identifying all regions corresponding to stars in the image, we obtain the centroid of each region by calculating a weighted average of that region,

$$x = \frac{\sum_{i=1}^{n} x_i \times I_i}{\sum_{i=1}^{n} I_i}, \tag{3.1}$$

$$y = \frac{\sum_{i=1}^{n} y_i \times I_i}{\sum_{i=1}^{n} I_i}, \tag{3.2}$$

where (x, y) are coordinates of the centroid, (x_i, y_i) are (x, y) coordinates of the i^{th} pixel in the region, I_i is intensity of the i^{th} pixel in the region, and n is the total number of pixels above the threshold value. The coordinates are described in the image plane coordinate system and are, thus, 2-dimensional.

There are various other ways of centroiding algorithms which improve the accuracy of centroiding and give a more definite central point location of a star region (A. Vyas, 2009). But these more compute intensive and necessary where accuracy of centroiding required would be very high. In the current case, a centroiding accuracy of about $(1/3)^{rd}$ of a pixel is sufficient as it corresponds to about $10''$ of centroiding error contributed in the net quaternion accuracy.

3.3.3 Geometric voting algorithm

Centroiding algorithm gives the coordinates of the stars in the image plane coordinate system. To identify stars, we employ the Geometric Voting algorithm (Kolomenkin, 2008), where the angles between each pair of stars detected in the image are matched to the angles between pairs of stars in the star catalog. We convert the star positions in the image plane coordinate system to a unit vector in the sensor body coordinate system using Eq. (3.3), where (u_x, u_y, u_z) are components of the unit vector of the star in the sensor body coordinate system, (x_u, y_u) are coordinates of the centroids in the image plane coordinate system, (x_c, y_c) are coordinates of the principal point (i.e. the point where the optical axis intersects the detector) in the image plane coordinate system, (pp_x, pp_y) are pixel sizes in x and y directions of the image sensor, respectively, and f_{mm} is the focal length of the imaging optics.

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \left(1 + \left((x_u - x_c)\frac{pp_x}{f_{mm}}\right)^2 + \left((y_u - y_c)\frac{pp_y}{f_{mm}}\right)^2\right)^{-\frac{1}{2}} \begin{bmatrix} (x_u - x_c)\frac{pp_x}{f_{mm}} \\ (y_u - y_c)\frac{pp_y}{f_{mm}} \\ 1 \end{bmatrix} \tag{3.3}$$

After conversion to unit vectors in sensor body coordinate system, the angle between each pair of imaged stars is obtained by taking dot products of their unit vectors as shown in eq 3.4. The pairs of image ID numbers and the angle between them are tabulated in an image angle-pair table.

$$\theta = cos^{-1}\left(\frac{\mathbf{a}.\mathbf{b}}{|a|.|b|}\right) \tag{3.4}$$

where **a** and **b** are unit vectors and $|a|$ and $|b|$ are their magnitudes which is unity.

A star catalog is divided into two parts: a unit vector list and a catalog angle-pair table. These lists/tables are generated using Hipparcos catalog of nearby stars as a base[2] by selecting stars brighter than a certain limiting

[2] available for download from http://www.heasarc.gsfc.nasa.gov

magnitude and calculating their unit vectors in the ECI coordinate system. The catalog ID of the star (HID), along with its unit vector, is tabulated in unit vector list. A catalog angle-pair table is generated by taking dot products of pairs of unit vectors, and HIDs of the pair of stars with their angular distance are tabulated. We keep only the entries that are smaller than the FOV of the optics. We sort the catalog angle-pair table in increasing order of the angle values. Both parts of the catalog are stored in the on-board flash memory, and are loaded into the RAM upon initializing the star sensor.

The Geometric Voting algorithm finds a match between each entry in the image angle-pair table with the catalog angle-pair table using a binary search method. All the entries in the catalog angle-pair table with angle values lying in a small range around the angle value in the image angle-pair table are selected. The imaged stars can possibly be any pair of stars from this selected star pairs. The HIDs of stars in the catalog angle-pair table cast a vote for the identified stars. This process goes on for all the entries in the image angle-pair table. At the end we find the maximum number of votes given for each imaged star. The HID corresponding to maximum number of votes is the most probable match. Further we start a verification process where the angle between each pair of stars with HIDs is verified to be lying in a small range around the angle between corresponding pair of stars in the image. This process verifies whether the identification of stars was done correctly.

3.3.4 Quaternion estimator (QUEST) algorithm

After identification of stars in the image, the last step is to calculate the quaternion of rotation between the ECI coordinate system and the sensor body coordinate system. This process is called attitude determination and can be implemented using various algorithms (Markley,F.L., et. al., 1979; Shuster, M. D., 2006, 1981). In our case, the output of the Geometric Voting algorithm gives the unit vectors of the stars in ECI coordinate system, and unit vectors in the sensor coordinate system are calculated from Eq. (3.3). These two set of unit vectors are inputs to the QUEST algorithm.

The rotation between any two coordinate systems can be described in different ways, such as rotation matrix, Euler angles, quaternions, etc. Each of these descriptions have their own applications and their own pros and cons. We select quaternions to describe the rotation/orientation, because this method is comparatively less computationally extensive.

The main aim of an attitude estimation algorithm is to use a pair of vector lists, one in ECI coordinate system and the other in sensor body coordinate system, to determine their relative rotation.

Let the unit vectors in ECI coordinate system be $\mathbf{v}_i$, and those in the sensor body coordinate system be $\mathbf{v}_b$. The equation $\mathbf{v}_b = R_{bi}\mathbf{v}_i$ holds true in an ideal case, where R_{bi} defines the rotation matrix between the ECI coordinate system and the sensor body coordinate system. Our problem is to identify R_{bi} from the unit vectors list obtained in the previous steps. One approach is a statistical approach, where we get many measurements of the $\mathbf{v}_b$ from the sensor body coordinate system in the form of star locations and identify those stars by Geometric Voting algorithm which gives $\mathbf{v}_i$ in the ECI coordinate system. We need to find R_{bi} which minimizes the loss function,

$$J(R_{bi}) = \frac{1}{2}\sum_{k=1}^{n} \omega_k \left|\mathbf{v}_{kb} - R_{bi}\mathbf{v}_{ki}\right|^2 \ , \tag{3.5}$$

where J is the loss function to be minimized, ω_k is the set of weights assigned to each vector pair measurement, and n is the number of correctly identified stars. This problem is called the Wahba's problem (Wahba, 1966) and different solutions for this problem are suggested in the form of different methods to calculate the attitude information. In an ideal case when all measurements are perfect, we obtain $J = 0$. However in practice, there will always be an error in measurement resulting in $J > 0$. The smaller J can be made, the better is the approximation of R_{bi}. We restate the loss function in terms of quaternions in such a way that it becomes an eigenvalue problem, where the largest eigenvalue is to be found. However, finding an eigenvalue is computationally very intensive for an embedded system. The QUEST algorithm was developed to bypass the expensive eigenvalue problem by approximating this process (Markley,F.L., et. al., 1979). The final form of the problem involves solving for **p** in the following equation

$$[(\lambda_{opt} + \sigma)\, I - S]\, \mathbf{p} = Z\,, \tag{3.6}$$

where

$$B = \sum_{k=1}^{n} \omega_k \left(v_{kb} v_{ki}^T \right) \;,$$

$$\lambda_{opt} = \sum \omega_k \;,$$

$\sigma = \mathrm{Tr}(B)$, i.e. sum of diagonal elements of B,

$$S = B + B^T \;,$$

$\mathbf{p}$ is the Rodriguez parameter which must be solved for ,

$$Z = \begin{bmatrix} B_{23} - B_{32} & B_{31} - B_{13} & B_{12} - B_{21} \end{bmatrix}^T \;.$$

Once the Rodriguez parameter p has been found, the attitude quaternion can be calculated using

$$\mathbf{q}_{quest} = \frac{1}{\sqrt{1 + \mathbf{p}^{\mathbf{T}}\mathbf{p}}} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \;, \tag{3.7}$$

where $\mathbf{q}_{quest} = \begin{bmatrix} \mathbf{q_1} \\ \mathbf{q_2} \\ \mathbf{q_3} \\ q_4 \end{bmatrix} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4.$

3.4 Simulations, performance and results

3.4.1 Centroiding algorithm

To simulate every step in this algorithm, we require an image of a star (reference image) with characteristics of the star sensor optics. We simulate this image by generating a Gaussian PSF with FWHM of 2 pixels. This image is

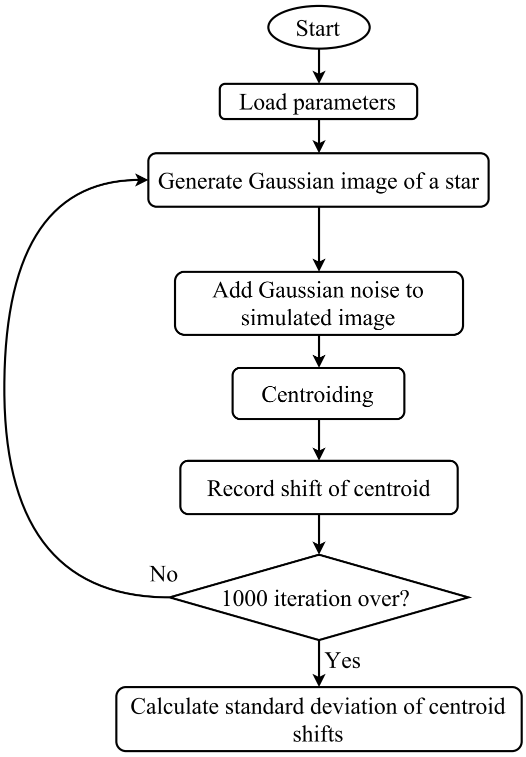

Start
Load parameters
Generate Gaussian image of a star
Add Gaussian noise to simulated image
Centroiding
Record shift of centroid
1000 iteration over?
No
Yes
Calculate standard deviation of centroid shifts

Figure 3.4: Flow of simulation of centroiding algorithm. This simulation is implemented over 1000 generated frames and the error in the centroid location calculated over the 1000 frames.

generated by a 2-dimensional function for a 1024 × 1024 pixel array,

$$G(x,y) = \frac{1}{\sigma_G\sqrt{2\pi}} \exp^{-\frac{(x-x_C)^2+(y-y_C)^2}{2\sigma_G^2}}, \tag{3.8}$$

$$\sigma_G = \frac{FWHM}{2\sqrt{2\log 2}}, \tag{3.9}$$

where:

x varies as $1, 2, \ldots 1024$;

y varies as $1, 2, \ldots 1024$;

(x_c, y_c) is the coordinate of the ideal center of the star in the image plane coordinate;

σ_G is the variance of the Gaussian function;

FWHM is the FWHM of the star image as expected from the *StarSense* lens.

Considering an image exposure time of 100 ms, we can estimate the noise introduced by the readout mechanism, dark noise and fixed pattern noise.

Table 3.1: Times recorded for image plane search (as simulated in MATLAB).

Pixel increment	Number of pixels checked	Time required
1 pixel	1048576	0.0956 sec
2 pixel	262144	0.0293 sec
4 pixel	65536	0.0102 sec

Knowing these characteristics and assuming various signal to noise ratio (SNR) for the star image, we have added Gaussian noise, matching the pixel noise in the image sensor, to the star image. The mean value of the Gaussian noise is the value of the dark signal, and its standard deviation σ is the sum of the readout and the dark noise.

3.4.1.1 Processing time

A significant fraction of the time of the centroiding algorithm is utilized in the image plane search process, and to reduce the time required in this process, we search only every alternate pixel. The time required to scan through the complete image reduces to almost $1/4^{th}$ of the default case. Sampling at every 4^{th} pixel speeds up the process even further, but that poses a possibility to miss the actual star because FWHM=2 pixel, by default. Times taken for each search methods are tabulated in Table 3.1. Simulations performed with different SNR suggest that we do not miss any star with SNR $> 3\,\sigma$ if we use the alternate pixel scanning.

3.4.1.2 Centroiding accuracy

We simulated an implementation of the complete centroiding algorithm in a sequence including image plane search, region growing/segmentation, and finally centroiding. This is performed on a reference image where, unlike in the ideal case, the calculated centroid has positional uncertainty due to added noise. The simulation is repeated 1000 times, each time generating a randomly varying Gaussian noise at each pixel, and the shift in the centroid is recorded for each image. Knowing the pixel scale of the instrument, the centroid error in angular shifts ($''$) can be calculated. The mean shift in these 1000 images gives a measure of the uncertainty of the centroiding algorithm due to the introduced noise. This uncertainty varies with the star magnitude and SNR as shown in Table 3.2.

Table 3.2: Variation of centroiding error with SNR of source

Magnitude	SNR	Centroid shift ($''$)
0	311.98	1.2
0.5	247.78	0.80
1	196.78	1.13
1.5	156.26	1.53
2	124.06	1.83
2.5	98.46	2.03
3	78.11	3.50
3.5	61.92	3.67
4	49.04	5.85
4.5	38.75	16.35
5	30.55	23.25
5.5	23.97	27.68
6	18.69	34.35
6.5	14.44	39.6

3.4.1.3 Memory requirement

The centroiding process occupies the greatest part of the total temporary storage memory (random access memory – RAM). All subsequent processing requires only the storage of the centroid coordinates during one iteration. We estimated the RAM requirements of the centroiding algorithm through basic calculations. The image sensor has a size of 1024 × 1024 pixels, with 10 bits per pixel. Thus, the memory requirement for one image is $1024 \times 1024 \times 10$ bits ≈ 1 MB. Apart from this, we have to store an image mask, which identifies the regions of the image that belong to a star, which accounts for another 1 MB of RAM. Furthermore, the image acquisition and image processing are run as two parallel tasks, which means that we need another 2 MB of RAM for the implementation. So much of RAM (4 MB) is not available on the FPGA chip and, therefore, we added an external RAM chip to the image processor PCB (Table 4.3).

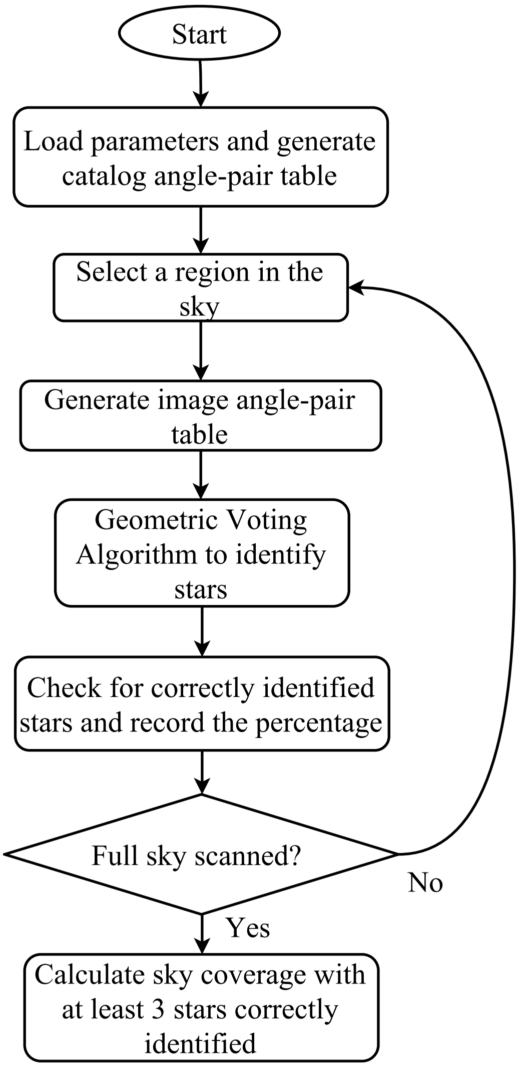

Start
Load parameters and generate catalog angle-pair table
Select a region in the sky
Generate image angle-pair table
Geometric Voting Algorithm to identify stars
Check for correctly identified stars and record the percentage
Full sky scanned?
No
Yes
Calculate sky coverage with at least 3 stars correctly identified

Figure 3.5: Flow of simulation of geometric voting algorithm. It is implemented by segmenting the whole visible sky into smaller overlapping fields as seen from the star sensor camera. Knowing the original identities of the stars, the identities found from the geometric voting algorithm are compared and a sky coverage obtained for a criteria of correctly identifying atleast 3 stars in the field.

Table 3.3: Number of entries in catalog angle–pair table

Limiting Magnitude	No. of stars	No. of entries
5.0	1608	11671
5.5	2819	35493
6.0	4995	108656
6.5	8789	332092

3.4.2 Geometric voting algorithm

The Geometric Voting algorithm is basically the implementation of a voting scheme. The inputs for this algorithm are the catalog angle–pair table and the image angle–pair table. The algorithm outputs the HID of the stars in the image. To evaluate the performance of the algorithm in terms of sky coverage and identification/verification accuracy, we simulate the ideal inputs required for the algorithm (i.e. the image angle–pair table) using physical parameters of the optics (FOV and limiting magnitude) and selecting sources from the Hipparcos star catalog.

3.4.2.1 Sky coverage

Here we verify the performance of the Geometric Voting algorithm in correctly identifying stars in any given field in the sky. We use a FOV of 10° and limiting magnitude of 6.5^m to model the optics. By varying the threshold in the centroiding procedure, we can adjust limiting magnitudes of detected stars in the image. We consider various limiting magnitudes from 5.0 to 6.5 in the analysis. The distribution of stars in the sky with different limiting magnitudes is shown in Fig. 2.4 in the previous chapter. The catalog angle–pair table is formed using these stars. Different limiting magnitude skies have different number of entries in the catalog angle–pair table depending on the number of stars in the sky (Table 3.3). The size of this catalog is taken into consideration while deciding on the flash memory size.

We divide the entire sky into 1728 overlapping fields, where each field is 10 deg (optics FOV) and fields are spaced at 5° from each other. From Hipparcos catalog, we find the number of stars in each field for different limiting

Table 3.4: Minimum and maximum number of stars in any field

Limiting Magnitude	Minimum no. of stars	Maximum no. of stars
5.0	1	18
5.5	1	27
6.0	1	39
6.5	4	70

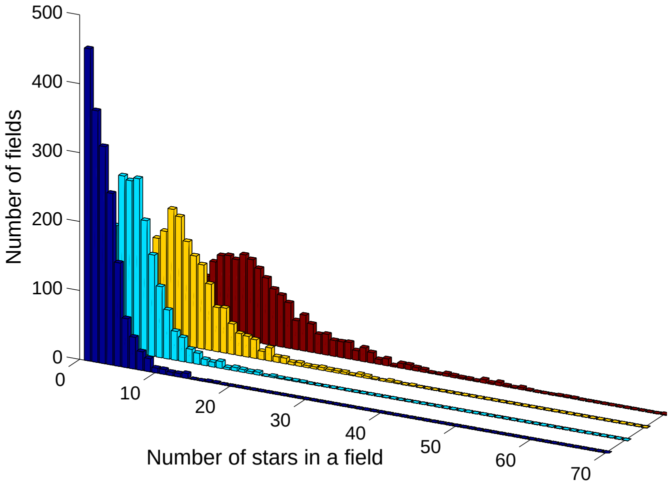


Figure 3.6: Histogram of the number of fields vs. the number of stars in a field. Blue, cyan, yellow and red are for limiting magnitudes 5.0, 5.5, 6.0 and 6.5, respectively.

magnitudes (Fig. 3.6). The minimum and maximum number of stars found in any field are tabulated (Table 3.4). For example, for a limiting magnitude of 5.0, the minimum number of stars in any field is 1. This shows that at any pointing of the telescope, there is always at least 1 star in the FOV.

Using the unit vectors of the stars in the selected field, we generate an image–angle pair table. The stars are renumbered in the order of their appearance in the selected field, keeping their HIDs in a separate variable. The Geometric Voting algorithm estimates the HID of a star in the selected field and verifies it by matching its angle in the image–angle pair table with the catalog–angle pair table. This process is repeated for every field. In Section 3.4.1.2, we have estimated centroiding accuracy of the optics for different magnitudes and corresponding SNR values, where the worst centroiding ac-

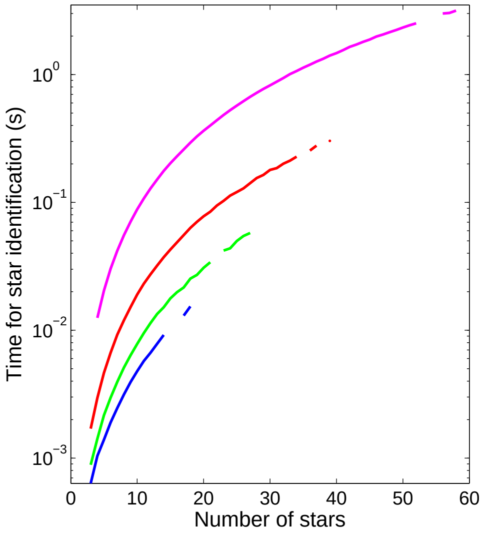


Figure 3.7: Time required for identification of stars with varying number of stars in field. Blue, green, red and magenta lines show the timings for limiting magnitudes 5.0, 5.5, 6.0 and 6.5, respectively

curacy is $\sim 35''$ for 6^m stars. The Geometric Voting algorithm searches for matching pairs in the catalog–angle pair table with a tolerance of $\sim 35''$. We introduce additional noise in image–angle pair table corresponding to the centroiding error arising in measuring the angles between the imaged stars, to simulate a realistic situation. We use the normal distribution with mean value of the worst-case of centroiding accuracy to simulate the noise to be added. The results obtained from this analysis are shown in Table 3.5.

Table 3.5: Results of Geometric Voting Algorithm Sky Coverage

	Limiting Magnitudes			
	5.0	5.5	6.0	6.5
No. of fields	910	1437	1704	1728
Correctly estimated	37.14%	64.02%	88.43%	98.95%
Correctly verified	37.14%	64.02%	88.43%	98.95%

3.4.2.2 Timing analysis

During the procedure of estimating the sky coverage, we also estimate the time required to identify the stars in the field. This time varies with the number of stars visible in the selected field. It is also affected by the catalog angle-pair table selected depending upon the limiting magnitude, because the number of entries in catalog angle-pair table is different for different limiting magnitudes (Table 3.3). The required times are shown in Fig. 3.7. Blue, green, red and magenta lines show the times for limiting magnitudes 5.0, 5.5, 6.0 and 6.5, respectively. It can be seen that as the number of stars in a certain field increases, the time required to identify them also increases significantly.

3.4.3 Quaternion estimator (QUEST) algorithm

The QUEST algorithm takes the unit vectors of reference points in the ECI and sensor body coordinate systems as inputs, and yields the estimated quaternions of rotation between the coordinate systems as output. To estimate the performance of this algorithm, we simulate the ideal inputs in a similar way as we did for the Geometric Voting algorithm. We already know the unit vectors of stars in ECI coordinate system visible in each field from the unit vector list of the star catalog as described in Section 3.3.3. We simulate their unit vectors in sensor body coordinate system by the following process. The rotation matrix between the ECI coordinates and the sensor body coordinates for a selected field is calculated from the following equation,

$$R_{bi} = \begin{bmatrix} \cos\alpha\cos\delta & -\cos\alpha\sin\delta & \sin\alpha \\ \sin\delta & \cos\delta & 0 \\ -\sin\alpha\cos\delta & \sin\alpha\sin\delta & \cos\alpha \end{bmatrix} \tag{3.10}$$

where α and δ are the RA and Dec of the center of the field the camera is looking at. The unit vectors in sensor body coordinate system of the stars visible in that field are the product of the rotation matrix with their unit vectors in the ECI coordinate system,

$$\mathbf{v}_b = R_{bi}\mathbf{v}_i\,, \tag{3.11}$$

where $\mathbf{v}_b$ is the unit vector of a particular star in the sensor body coordinate system, and $\mathbf{v}_i$ is the unit vector of that same star in the ECI coordinate

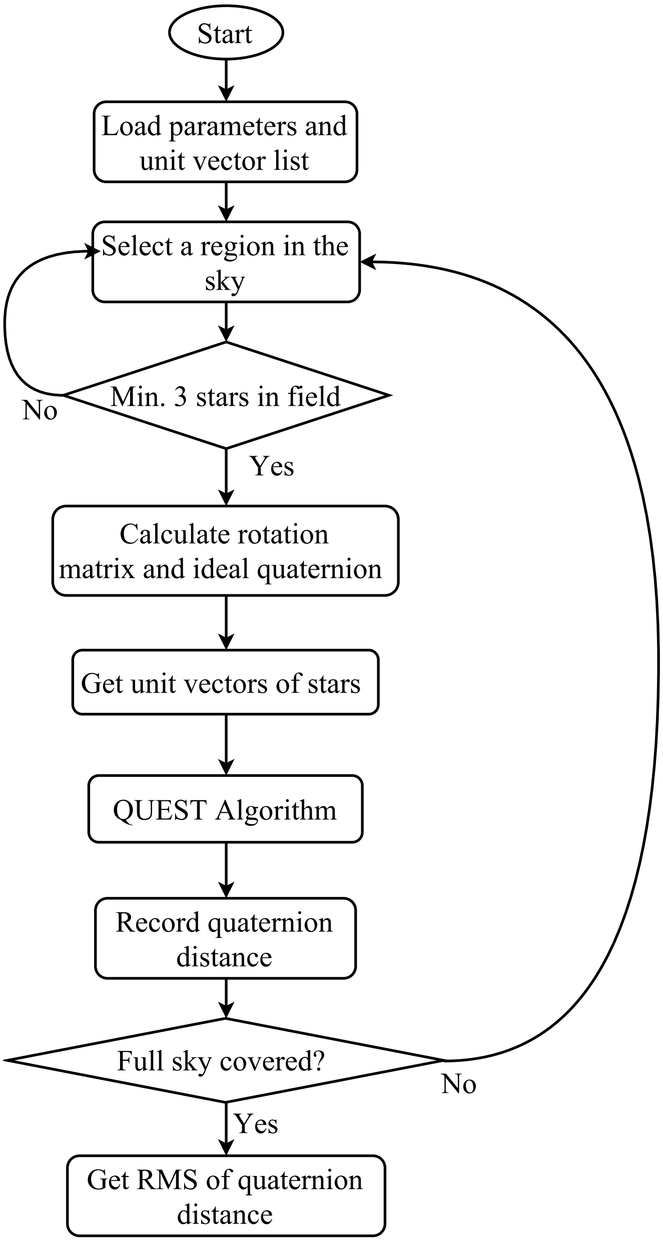

Start
Load parameters and unit vector list
Select a region in the sky
Min. 3 stars in field
No
Yes
Calculate rotation matrix and ideal quaternion
Get unit vectors of stars
QUEST Algorithm
Record quaternion distance
Full sky covered?
No
Yes
Get RMS of quaternion distance

Figure 3.8: Flow of simulation of QUEST algorithm. The RMS quaternion distance between the computed quaternion and the reference quaternion is obtained for all the possible fields in the sky obtained after segmentation.

system. The ideal quaternion is obtained from R_{bi} (Eq. 3.10) by

$$\mathbf{q}_{ideal} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \dfrac{m_{21} - m_{12}}{2\sqrt{1 + m_{00} + m_{11} + m_{22}}} \\ \\ \dfrac{m_{02} - m_{20}}{2\sqrt{1 + m_{00} + m_{11} + m_{22}}} \\ \\ \dfrac{m_{10} - m_{01}}{2\sqrt{1 + m_{00} + m_{11} + m_{22}}} \\ \\ \dfrac{\sqrt{1 + m_{00} + m_{11} + m_{22}}}{2} \end{bmatrix}, \tag{3.12}$$

where m_{00}, m_{01}, m_{02}, m_{10},... m_{22} are the elements of the rotation matrix. The output quaternion from the algorithm is then compared with this quaternion by calculating the distance between these quaternions using their quaternion dot product

$$d = cos^{-1}(2\mathbf{q}_{ideal}.\mathbf{q}_{quest} - 1) \tag{3.13}$$

3.4.3.1 Sky coverage and accuracy

For quaternion calculation we have selected a criterion of having minimum of 3 stars in any given field. Therefore, the number of fields in the sky with more than 3 stars in a field defines the sky coverage for the QUEST algorithm. The quaternion distance (Eq. 3.13) is calculated for all the fields in the sky. Its standard deviation (RMS error) gives the accuracy of the QUEST algorithm, which is almost independent on limiting magnitude and very small, of the order $\sim 10^{-7}$ deg. The number of fields in the sky with more than 3 stars in a field are tabulated in Table 3.6.

Table 3.6: Performance of QUEST for different limiting magnitudes

Limiting Magnitude	Sky Coverage
5.0	52.66%
5.5	83.16%
6.0	98.61%
6.5	100%

Table 3.7: Estimated parameters from the software package

Parameter	Value
Centroiding accuracy	$35''$
Distortion residual	$8''$
Centroiding time	0.05 sec
QUEST numerical error	$\sim 10^{-7''}$
Realtime memory requirement	$\sim 4MB$
Catalog size	$\sim 1.5MB$
Sky Coverage	90%

3.5 Conclusion

We have put together a complete software package to reduce star sensor images to quaternions and to evaluate the performance of the operational algorithms. The package simulates centroiding, Geometric Voting and QUEST algorithms, and evaluates such performance parameters as attitude accuracy, calculation time, required memory, star catalog size, sky coverage, etc., and estimated the errors introduced by each algorithm. The testing is parametrized for different hardware parameters of the star sensor, such as the focal length of the imaging setup, FOV of the camera, angle measurement accuracy, distortion effects, and others. We conclude with the following remarks:

- For *StarSense*, we find that a limiting magnitude of V=6.0 is optimal to get significant sky coverage with minimal calculation time.
- This software package is robust, fast, user-friendly in terms of varying hardware parameters, and easily portable to various operating platforms.
- Due to the parametrized approach in package development, it can be applied to evaluate the performance of such algorithms in any star sensor.

The estimated performance parameters and the estimated errors are tabulated in Table 3.7. The source codes of the software package can be obtained from a github repository (Mayuresh Sarpotdar, 2016).

Chapter 4

Design and analysis of *StarSense*

4.1 Introduction

Following the star simulation and software analysis as described in the earlier chapters, the desired operational parameters of *StarSense* hardware were determined for *StarSense* . Various physical parameters viz. field of view, pixel scale, memory size for catalog and real-time image processing, etc. of the *StarSense* were obtained from the analysis. In this chapter we describe the design process of the constituent components of the *StarSense* . The simulations and analysis done to arrive at the respective designs of the components are also described in this chapter.

There are other star sensors commercially available in the space industry. (For example: Astro 15(Jena Optronik , 2015), Leonardo A-STR(Leonardo

Table 4.1: Design requirements for the *StarSense*

Accuracy	$\sim 30''$
Weight	$\sim 500gms$
Update rate	$\sim 5Hz$
Lifetime	$\sim 1year$
Power consumption	$< 5W$

Table 4.2: *StarSense* Camera Parameters as derived from general sensor requirements

Field of View	10° circular FOV
Focal Length	80 mm
Aperture	F/2.6
Limiting Magnitude	$V_m = 6^m$
Image size	1024×1024
Pixel size	$15 \times 15\ \mu\text{m}^2$
Pixel scale	$36''$/px
Integration time	100 ms
ADC resolution	10 bit
Weight	600 gm
Power	2 W

A-STR, 2016), Sodern Hydra(Sodern Hydra, 2017) etc.) A comparative study of their capabilities, cost and characteristic properties revealed a possibility of further improvement of these properties to suit them to small satellite applications. Most of these sensors are designed for heavier class satellites which can accommodate weight budget of such sensors to be of the order of few kgs and a power budget of tens of Watts. But considering the modern paradigm of nano and micro satellites and high altitude balloon platform based experiments, such large weight and power margin are not acceptable. On the other hand, the accuracy requirements for these new paradigm payloads is not as constrained as the large satellites/ experiments. This means a slightly degraded accuracy is acceptable for the new age payloads with a constraint on the mass and power budget. Considering all these factors and the analysis as shown previously, a concise requirement from the sensor was arrived at. This included a major design goal to make a compact, light weight, moderately accurate star sensor to use in the pointing control system for high altitude balloon observatory described in chapter 1. These requirements are tabulated in Table 4.1

Using these requirements as baseline and from the understanding from the previous chapters, the camera parameters were obtained. Table. 4.2 and 4.3 summarizes the required hardware parameters for the camera.

Table 4.3: *StarSense* Specifications for the electronics subsystem derived from general sensor requirements

Detector	Star 1000 (Radiation-hardened)
Readout	MIL Grade Spartan-6 FPGA (XQ6SLX150T)
Non-volatile memory	8 MB (For star catalog and bit stream)
RAM	64 MB (For on-board image processing)
Operating Voltage	5 V

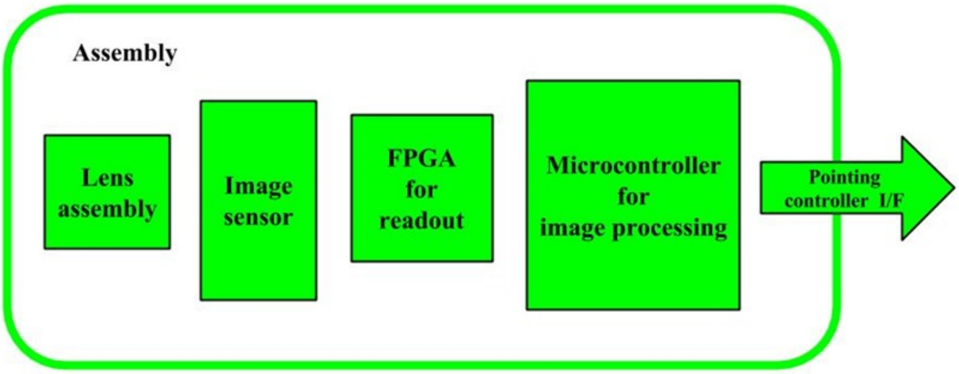


Figure 4.1: Hardware block diagram of star sensor based on sensor requirements.

4.2 System block diagram

StarSense essentially is a wide field of view camera which takes pictures of the star field, detects stars, identifies them using a star catalog stored on-board and calculates a rotation quaternion between the sensor coordinate system with respect to the earth centered inertial coordinate system which is used as the reference coordinate system. We visualized a typical block diagram of a wide field camera as shown in Fig. 4.1. The lens forms an image of the star field on the detector. An FPGA based embedded system is used to readout the image captured by the image sensor and to store the image in on-board memory for real-time processing. A microcontroller processes the image in real-time and generates the quaternion. The quaternion is then communicated to the satellite on-board computer (OBC) through a standard satellite interface.

We followed a detector-centric process for the design of the *StarSense* optics, mainly because of the long procurement time for the detector. We used the image sensor Star 1000 (NOIS1SM1000A) from On Semiconductors. It is a

1 Megapixel detector in a $1k \times 1k$ form factor. This detector is space qualified and radiation hardened. To design the optics the preliminary variables are: integration time, field of view of the optics and the collecting area (entrance aperture). We targeted a data output rate of 5 Hz which limits the total image capture time to about 200ms of which maximum integration time can be 100ms and the remaining 100ms for sequential calculations. From previous chapter conclusions we know that the field of view required for the desired sky coverage should be $\sim 10°$. From the same sky simulation using the star catalog, we concluded that the required limiting magnitude of the system is 6^m. From these requirements, we derive the optimum size of entrance aperture of optics considering weight of the lens. From the noise characteristics as defined in the detector data sheet viz. readout noise, dark noise, fixed pattern noise etc. we can estimate the collecting area required for detecting 6^m stars for a desired signal to noise ratio for detection. Further a lens is designed to minimize off axis aberrations like coma and distortion, spherical aberration and chromatic aberration. In addition, the lens has to be immune to the vibration that it undergoes during the launch and maintain the same image quality in space after separation, in spite of the daily temperature cycling which it would undergo while in an orbit in case of a satellite.

To capture the images from the detector, an electronic system was designed. The electronics was segregated into 2 pcbs, the detector pcb and image processor pcb. The image processor is based on a Spartan-6 MIL grade FPGA. Spartan-6 features a hardware memory controller block which is key to many image processing and video processing operations implemented on FPGAs. We implement a multi-port memory controller (MPMC) design based on this memory controller block to share the RAM between the image sensor readout logic implemented on the FPGA and the microcontroller which is used to carry out the real time image processing.

The optics and the electronics being the main functional components of the star sensor, once they are designed and finalized, a housing is designed to hold all of these together. A detailed analysis of the housing to simulate its durability against vibrations recommended for PSLV launch loads is done. Also a suitable baffle to avoid stray light from the Sun or the Earth (albedo) is designed considering the required Sun and Earth avoidance angles. We describe in detail, the design process and analysis further in this chapter.

4.3 Optical design

4.3.1 Lens design

Lens being the most crucial part of the sensor was designed first. From the required specifications of the lens, a basic optical design was arrived at. The process of designing the lens in-house, carrying out tolerance analysis on it, getting it fabricated had an advantage of the knowledge of various lens characteristics. Various constraints were considered such as weight, a condition of seeing a minimum of 3 stars in FOV in any field of the sky, limiting magnitude etc. As mentioned earlier, we followed a detector centric design process and hence the properties of the detector were considered while designing the lens. The image sensor is the only electronic component directly open to the radiation environment and, therefore, its performance is prone to degradation with time. We used a radiation-hardened CMOS detector Star-1000, which is sensitive to low light conditions and has a high dynamic range. The size of each pixel of this detector is $15\mu m \times 15\mu m$. The optics was designed, so that the point spread function (PSF) is maintained at a full width at half maximum (FWHM) of 2 pixels, which corresponds to a star spread of 4×4 pixels, at all field positions i.e. up to the farthest off-axis point of $\sim 5^{\circ}$. The lens assembly was designed to sustain vibrations experienced during the satellite launch using specially designed spacers between lens elements. A baffle was designed to be used to prevent the stray-light from the Sun and the Earth entering the optical system, thereby contributing to the sky background. In operational conditions the equivalent heat generated by an image of the sun on the detector was estimated and it was realised that in cases where the slew rate is very slow, the image of sun could potentially damage the detector due to excessive heating. Hence, we have also used a thermal cut-down filter (a hot mirror), which reflects the IR and UV wavelengths and only allows optical wavelengths from 450–750 nm to pass through, to prevent the detector from excessive heating by direct Sun.

Further to design the lens, the aperture size of the lens was to be determined. This was obtained using a signal to noise analysis. This was done using the detector noise information. We considered a tessar lens system (Milton Laikin , 2006) as baseline to design this optics. A tessar lens system consists of 2 singlet lenses and a doublet lens in a Cooke triplet fashion. Therefore there are 6 surfaces where light reflection occurs. A transmission of 0.9 was considered at each of these surfaces, giving a net transmission of about 0.65. This transmission efficiency, random noise data and known integration time for each image was considered to arrive at the signal to noise ratio analysis. The variation of SNR with aperture diameter is shown in Fig. 4.2. This analysis also provided a way to visualize the variation of signal collected by the image sensor for the desired limiting magnitude star at varying diameters of entrance apertures. The obtained parameters from this simulation are tabulated in Table 4.4.

Table 4.4: Requirements and desired characteristics of the lens

Min. stars in any field	5 stars
Field of view	10°
Focal length	86 mm
Entrance aperture	30 mm
Limiting magnitude	6^m at 3σ level

We designed a lens system for compactness and ease of procurement. A fixed focal length lens ensures robust imaging setup. The optics design is shown in Fig. 4.3 and Table 4.5. We have selected a Tessar lens system as baseline which is a 4-element design with a stop after the second element and cemented third and fourth elements forming a doublet. This arrangement provides 13 independent variables (7 radii of curvatures, 4 element thicknesses and 2 inter-element distances) which are sufficient to control 7 primary aberrations (5

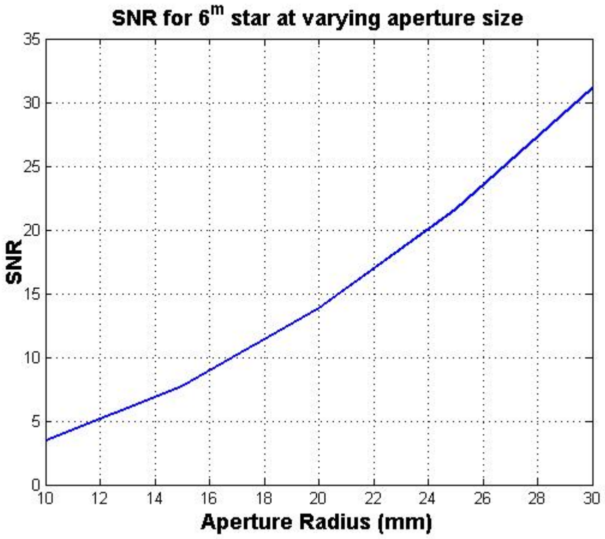


Figure 4.2: Signal to noise ratio for detecting a 6^m star at different optics aperture diameter. The figure gives a ballpark estimation of the signal to noise ratio with different aperture radii. It can be noted that the variation is in square proportion of the radii.

third order Seidel aberrations and 2 first order chromatic aberrations). The glasses are selected from the CDGM catalog (one of the catalog from a glass manufacturing company named CDGM) due to their quick availability and satisfactory specifications. A suitable combination of crown and flint glasses are selected to reduce chromatic aberrations. Performance analysis of the lens design is carried out using an optical design software called ZEMAX. The metric used to determine the image quality of the lens system is called the spot diagram or the point spread function of the lens. Another useful parameter to determine the image quality is the encircled energy at the image plane inside a circle of varying radius from the center point determined by the chief ray intersecting the image plane. This spot diagram and encircled energy diagram are obtained from the ZEMAX software. The variables as mentioned previously are to be adjusted so as to achieve a spot diagram spreaded over 4 pixel area on the image sensor i.e. RMS radius of the spot diagram should be $\sim 30\mu m$ considering the $15\mu m$ pixel size of the detector. In terms of encircled energy atleast 50% of the encircled energy should be within the central pixel. These spot diagrams and encircled energy for the optimized lens configuration

Table 4.5: Lens design data for *StarSense* lens system

Surf	Radius	Thickness	Glass	Semi diameter
OBJ	Infinity	Infinity	-	-
1	61.6251	8.00	H-K9L	15.00
2	323.7138	10.00	-	15.00
3	-43.8124	8.00	H-F13	15.00
4	-58.4967	5.00	-	15.00
STO	Infinity	5.00	-	14.0896
6	90.9504	4.00	H-F13	15.00
7	26.8273	0	-	15.00
8	26.8273	8.00	H-K9L	15.00
9	-70.9465	65.3430	-	15.00
IMA	Infinity	-	-	7.5

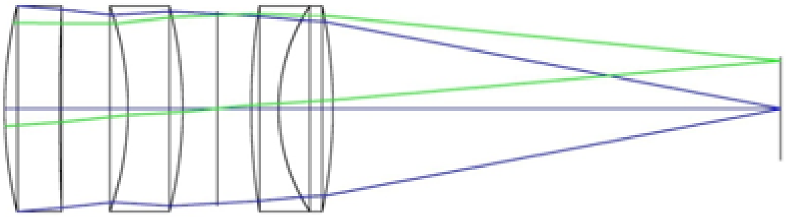


Figure 4.3: Visualisation of the designed lens system. It consists of 3 lens elements 2 of them are singlets and one is a doublet. The general layout is of a modified cooke triplet or a tessar lens system.

to achieve the image quality requirement are shown in Fig. 4.4 & Fig. 4.5 respectively.

In addition to image quality effects, distortion significantly affects the star sensor operation. The star matching algorithm relies on the correct measurement of angles between imaged stars. Due to distortion effect, position of stars on the image plane shifts, causing erroneous measurements of angles between them. This reflects in error in attitude estimation by the algorithm. The distortion graph shows that the maximum distortion introduced in the image due to the optics is only 0.1% which amounts to roughly 8$''$ considering the pixel scale. Moreover it occurs at the edges of the field and is less in the near axis fields.

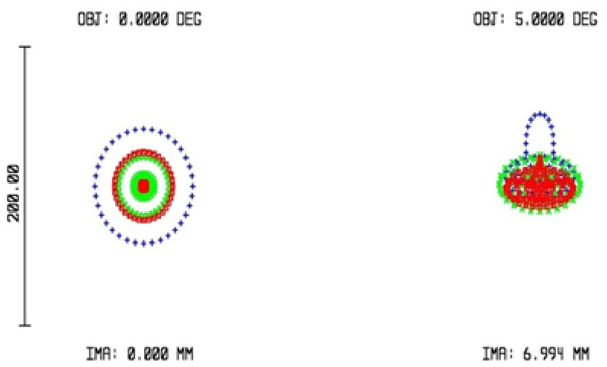


Figure 4.4: Effective spot diagram for the designed lens system

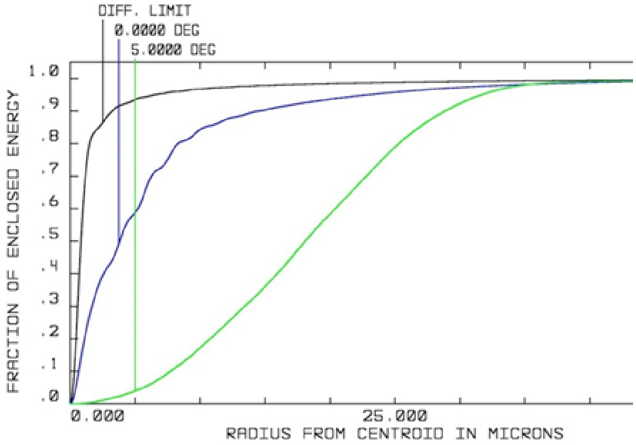


Figure 4.5: Encircled energy diagram for the designed lens system

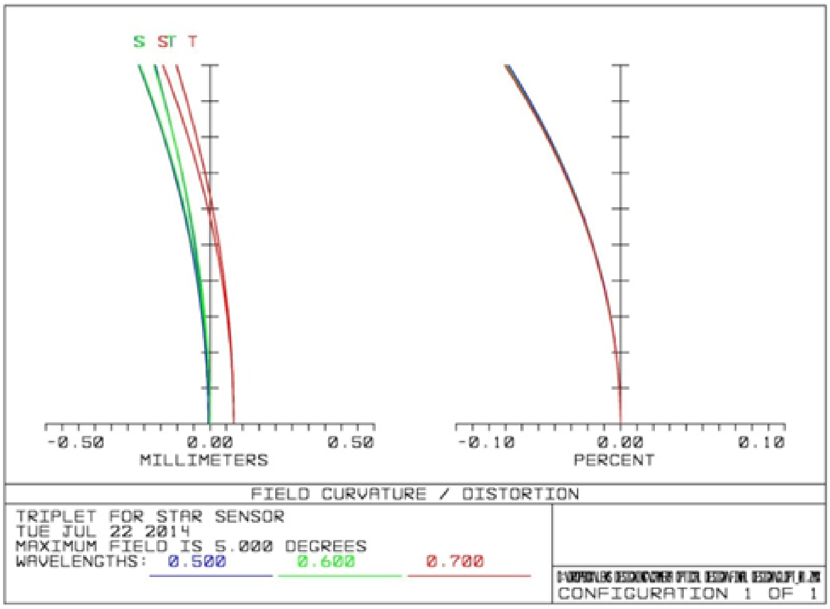


Figure 4.6: Field curvature and distortion estimates for the designed lens system

4.3.2 Lens assembly design

After having fabricated the custom designed lenses for the lens system, holding them in the desired configuration with the determined mechanical tolerances is very important. A tolerance analysis was carried out on the inter element distances to determine the variation in the focal length with variation in the distances. An assembly was designed to hold these lenses in place and make the system easy to mount in an independent camera system. Figure 4.7 shows the assembly used for holding the lens elements together in a group. A fixed assembly is used to avoid disturbances in lens positions because of vibrations. A silicone rubber gasket is used to fix the lens assembly in the housing.

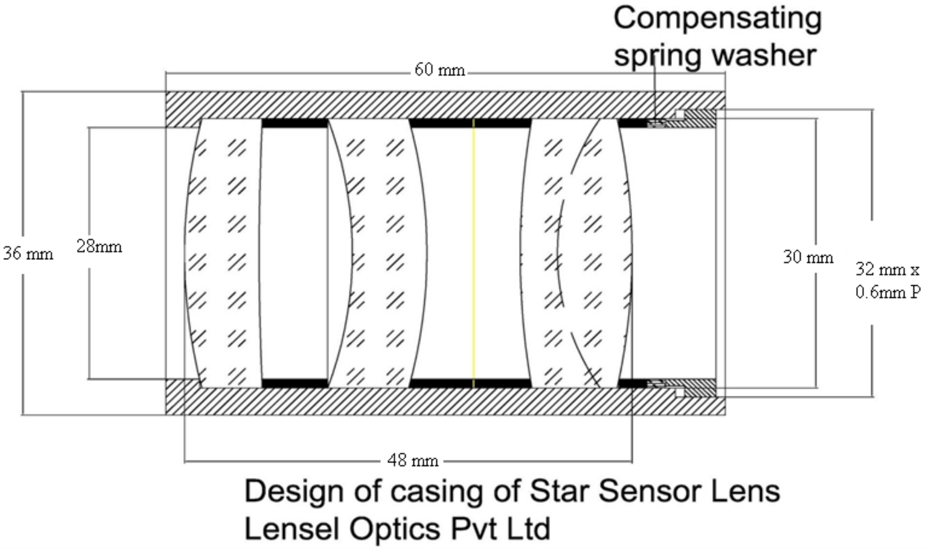

Compensating spring washer
60 mm
36 mm
28mm
30 mm
32 mm x 0.6mm P
48 mm
Design of casing of Star Sensor Lens
Lensel Optics Pvt Ltd

Figure 4.7: Design of a lens assembly and the housing to hold the lenses fixed at their desired places. A compensating spring washer is used to maintain a preload on the lens surfaces at the point of contacts.

4.3.3 Detector

We used the Star 1000 CMOS image sensor from On Semiconductors as mentioned in the previous section. This is a radiation hardened sensor that has been used in many space missions and has huge space heritage. The main features of the detector are as tabulated in Table 4.6. The maximum integration time for the detector is limited by the star sensor data update rate

Detector	Star1000 (NOIS1SM1000A)
Resolution	1024x1024 (1Mpixel)
Image sensor size	15mmx15mm
Pixel size	$15\mu m \times 15\mu m$
ADC resolution	10 bits
Spectral range	440nm - 1100nm
Power consumption	400mW

Table 4.6: Star1000 detector specifications

criterion and is assumed to be 100 ms. This detector was selected because of the following unique desirable features:

- Rolling shutter and region of interest readout capability: Useful for fast data acquisition in the tracking mode of the star sensor.
- Radiation hardening: Gives long life time for the detector.
- High sensitivity: In lower exposure times even fainter stars can be detected.

This image sensor is available in different combinations of filters, packages and qualification level formats. We used a monochrome image sensor because, a net estimate of the star magnitude was more useful than the actual color or color index of the star. It features on-chip Fixed Pattern Noise (FPN) correction, a programmable gain amplifier, and a 10-bit Analog-to-Digital Converter (ADC). The photons incident on each pixel of the image sensor generate electrons in those pixels. These electrons are then converted to voltages which are in turn converted to digital values using the on-chip ADC. A block diagram of the internal architecture of the image sensor is shown in Fig. 4.13. The addressing of each pixel is achieved using column and row decoders present in the image sensor. The charge to voltage conversion happens at pixel level in CMOS image sensors. A programmable gain amplifier allows to adjust the gain of the image sensor and amplifies the output voltage generated by the pixels. The 10-bit ADC lies at the end of the analog chain. All of these processes to happen sequentially for every pixel in the detector requires an external control signal generating system. This is called the readout controller.

4.4 Electronics design

4.4.1 Introduction

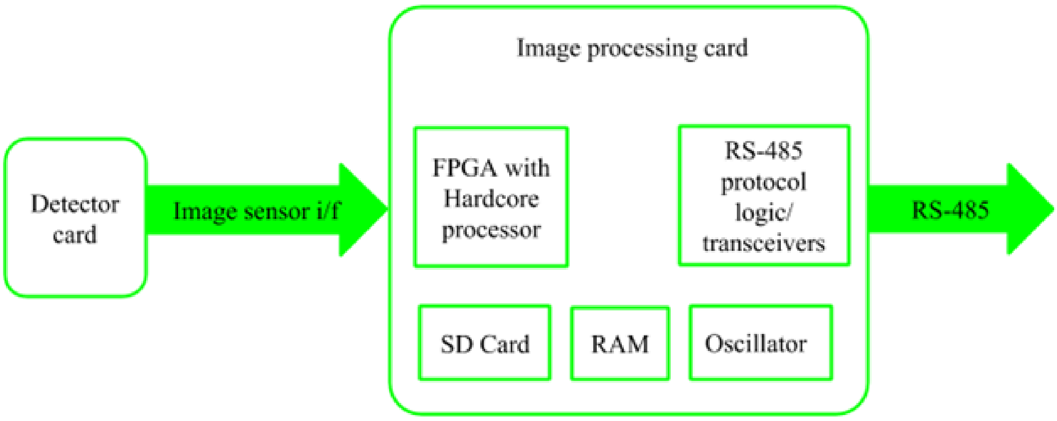


Figure 4.8: Block diagram of electronics subsystem for the star sensor. The overall system is split into two separate cards: detector card and image processing card. The detector card essentially only includes the detector used. This diagram shows the contents of the image processing card

The electronics subsystem constitutes a core sub system of the *StarSense*, because all the image acquisition, processing and output generation is done in this system. The readout controller generating the clock signals to coordinate the data acquisition from the image sensor operates at a frequency of 10 MHz to read the data from the image sensor at a rate of roughly 10 frames/second. To implement these clock signals a dedicated digital circuit based electronic system is necessary. Also, the data generated in payloads on high-altitude balloons or in small satellites is large in size. To download this data through the limited down link capability available on a satellite or a high-altitude balloon, the on-board processing of the data is required to compress it heavily. We have developed a field programmable gate array (FPGA)-based application board which can be used as a readout controller for various image sensors as well as for real-time image processing of the captured data. An FPGA is a programmable dedicated digital circuit which can be used to implement various digital interfaces, glue logic etc to interface with sensors and process their data in real-time. These utilities can be implemented in various digital interfaces without major changes in the hardware. Hence, it is a general purpose image processing board.

For any image processing board, the essential constituent components are a controller to acquire images from the detector, RAM for temporary storage of the images, a microprocessor for processing the image data, and a permanent memory for on-board storage of catalog or reference data. Our FPGA board combines all of these requirements. This board can be used to read the digital output from any standard semiconductor detector, e.g. CMOS or CCD. The acquired image is sent to RAM, and is processed by the microprocessor embedded in the FPGA. The processed image is stored in a non-volatile memory (SD card), and can be either transmitted to ground station from space, or in case of high altitude balloons be retrieved after landing.

The electronics system is divided into two separate printed circuit boards (PCBs): detector PCB and image processor PCB. The image sensor and its biasing circuits are mounted on the detector PCB, which is fixed inside the star sensor structure at the focal plane of the *StarSense* optics. The detector is connected to the image processor board through a flat plastic ribbon cable. Such an arrangement allows for the image processor board to be removed from the structure without disturbing the focal plane alignment of the detector PCB. The readout system for the image sensor is implemented using a MIL-Grade Spartan-6 FPGA on the image processor PCB. This PCB also hosts SDRAM, required for the online processing of images, and a flash memory for nonvolatile storage of the bitstream and a star catalog. Refer StarSense Image processor board user guide in appendix for more details about schematics of the board and other useful details.

4.4.2 FPGA board block diagram

The main components of the FPGA board and their layout are shown in Fig. 4.9 and their specifications are shown in Table 4.7. An FPGA is an integrated circuit designed to be configured by a programmer, or end user, to carry out a specific set of tasks. An FPGA contains programmable logic components called logic blocks, and a hierarchy of re-configurable interconnects that allow the blocks to be wired together to form different configurations. We have selected a MIL-grade Spartan-6Q FPGA from Xilinx[1] because of its wide operating temperature range. The configuration program for an FPGA is known as a bit-stream. The FPGA consists of configurable logic blocks where the configuration is volatile, which means that the FPGA has to be reprogrammed at each power cycle. We have selected a Master Serial programming interface for the FPGA, wherein the FPGA reads its bit-stream from a flash memory on its own using a serial transfer protocol, called serial peripheral interface (SPI), and configures itself. The size of the bit-stream determines the required size of the flash memory chip. In our case, for a Spartan-6Q FPGA device, the bit-stream size is 4 MB. Other than the bit-stream, the flash memory can be used to store data that should be permanently available to the FPGA e.g. a star catalog, some configuration parameters for the payload, etc. Therefore we have selected a flash memory chip of 8 MB. The image sensors which we have used has a size of 1024 × 1024 pixels, each pixel with a 10-bit digital value. This corresponds to an image size of ~ 1.2 MB being processed by the FPGA every 100 ms. We have selected a 64 MB RAM, which is sufficient for image processing tasks on images from the selected detectors. For the permanent storage of processed data, we use an SD card of 32 GB. The FPGA board can be connected to the on-board computer (OBC) of a satellite, or a balloon payload, through the standard RS485 protocol. During the development cycle, the FPGA can be connected to a computer through a USB port.

4.4.2.1 Voltage regulators

The FPGA board can be powered by an external computer (a laptop, a satellite/balloon OBC) through either a USB cable or through a DB-9 (9 pins) connector (Fig. 4.1). In addition to this, it can also be powered by a DC jack provided on the board. The FPGA needs 3 different voltages: 3.3 V, 2.5 V

[1] http://www.xilinx.com

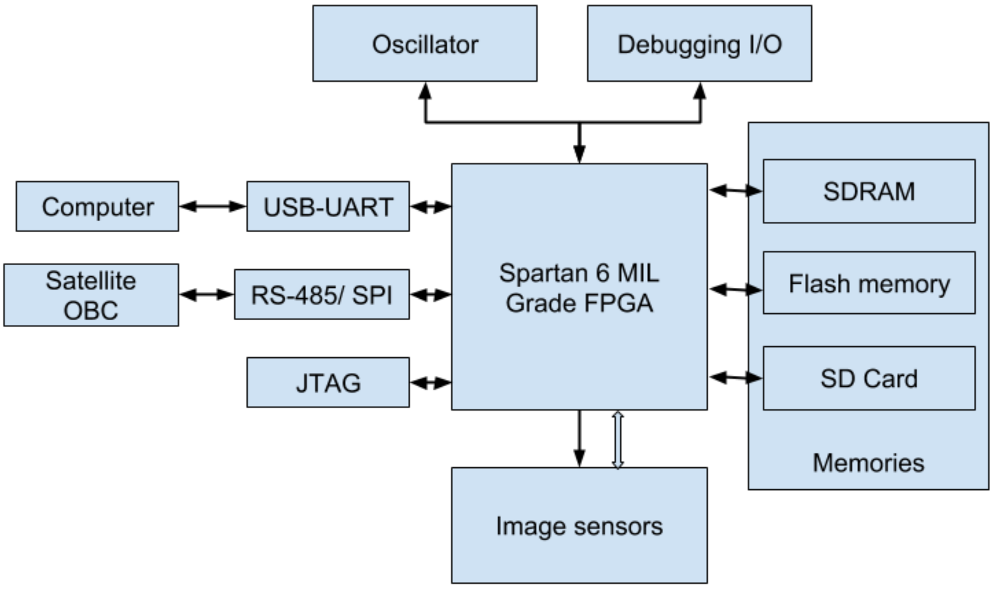


Figure 4.9: Block diagram of the FPGA board. It shows how different components present on the board are connected to the Spartan-6 FPGA. Please refer Appendix-A to know more about the schematics.

Table 4.7: Technical Specifications of the FPGA board

Component	**Specifications**
FPGA	Spartan-6Q (XQ6SLX150T-2FGG484)
Size (mm)	65 × 65
Weight	50 gms
Power	3 W
RAM†	64 MB SDRAM (MT46V32M16)
Flash memory†	8 MB chip (N25Q064A)
SD card	32 GB
Connectivity	RS485 (MAX481), USB-to-UART (FT232)

†

Micron Technology, Inc., USA. https://www.micron.com/

and 1.2 V. It needs 3.3 V for general-purpose input/output pins, image sensor interface, SD card, flash memory interface, and RS485 interface. Voltage of 2.5 V is needed to connect to the SDRAM. Finally, it needs 1.2 V for its internal functioning. All these voltages are generated from the unregulated 5 V input through the voltage regulators. The regulated voltage output should be switched on in a sequence to ensure successful programming of the FPGA. The FPGA is set in master serial configuration mode, and therefore, the flash mem-

Regulators	Configuration 1	Configuration 2
3.3V regulator	66%	66%
2.5V regulator	75%	50%
1.2V regulator	48%	24%

Table 4.8: Operating efficiency of regulators in various configurations

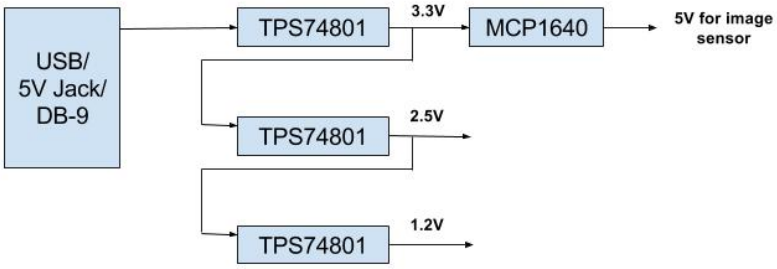


Figure 4.10: Voltage regulators configuration 1 for the FPGA board

ory chip should be powered on and ready to accept memory read commands, before the FPGA starts sending them. Therefore, the voltage regulators are connected in a chain as shown in figure (Configuration 1 - Fig. 4.10). The programmable soft-start feature of the voltage regulator ensures that the flash memory is powered on at least 3 ms prior to the FPGA, and is ready to accept read commands from the FPGA. The image sensors require a 5 V supply, which can be switched on or off in order to reduce power consumption, when not in use. We do not use the 5 V input directly, instead a boost regulator is used to increase the voltage from 3.3 to 5 V. Such configuration gives a stable voltage for the image sensors, which is a critical part of the circuit. There could be another topology of the regulator chains possible in the design (Configuration 2 - Fig. 4.11), wherein the 3.3V, 2.5V and 1.2V could be generated from an input voltage of 5V. This would be a parallel topology of the regulators. The 5V supply for the image sensor could be given from the 3.3V supply.

The efficiency of a linear regulator is proportional to the ratio of the output voltage to the input voltage of the regulator. Thus a comparison of the linear regulator operating efficiency can be tabulated as shown in Table 4.8. Hence we preferred the first configuration to achieve higher efficiency of regulators.

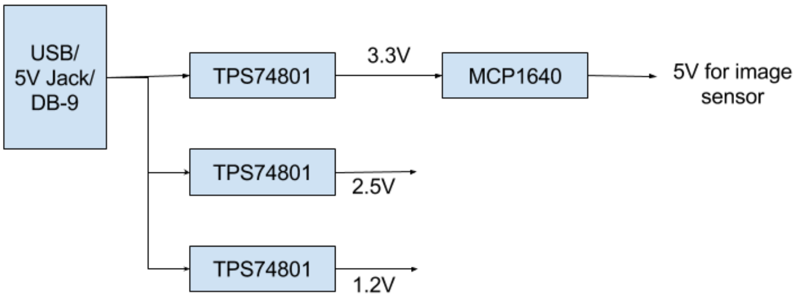


Figure 4.11: Voltage regulators configuration 2 for the FPGA board

4.4.2.2 Oscillator

The circuit has to operate in a wide temperature range ($-40°C$ to $85°C$). Hence, most of the components selected are industrial grade or MIL grade. Various parameters of the circuit components vary with temperature affecting the operation of the circuit. One of the most affected parameter is the operating frequency of the circuit. An accurate timing signal is provided to the circuit using a CMOS oscillator component in the circuit. The frequency of an oscillator drifts with temperature. This would directly affect many circuit operating parameters in our case viz. integration time, baud rate of serial interface between the sensor and the OBC etc. and thus the circuit would fail to operate in the desired way at different operating temperatures.

A special oscillator called Temperature Controlled Crystal Oscillator which has a temperature compensating circuitry inside along with the crystal is used in our circuit. This additional circuitry changes the frequency of oscillation with temperature in exactly opposite way as the crystal inherently does. Thus, the effect of frequency variation of the crystal output is nullified by the compensating circuitry. A normal crystal is available with specs of $\sim \pm 50ppm$ frequency variation about the nominal value, whereas a TCXO is available with specs of $\sim \pm 5ppm$ frequency variation about the nominal value. We are using one such TCXO on the FPGA board in the application circuit. This avoids the necessity of active temperature control system for the electronics PCBs.

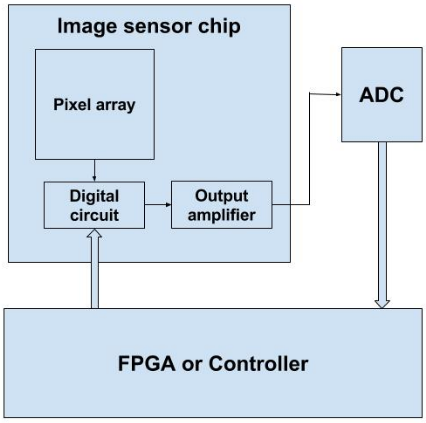


Figure 4.12: General schematics of components involved in a readout process of an image sensor. The host controller generates various control signals to operate the image sensor chip. The values read from the pixel array are output from a charge to voltage amplifier with a programmable gain output amplifier to an on-chip or off-chip ADC. The digital output from the ADC is then collected by the host controller.

4.4.3 Readout process

An image sensor is essentially an array of photo diodes (pixels) which measure the light intensity at each pixel. The output of each pixel is an analog voltage. Each pixel is connected to an output amplifier on the chip through digital circuitry. The clock signals generated by the FPGA board, control this digital circuitry and manage the sequence of connections. The analog voltage of the output amplifier is converted to a digital value by an ADC. This ADC value is also sampled by the same FPGA board. Figure 4.12 gives a general outline of this process. The process of collecting digital data (a measure of the light intensity on each pixel) is called the readout process.

4.4.3.1 Image sensor internal architecture

To read the pixel data from the image sensor we have to generate the clock signals for various internal modules of the image sensor from the FPGA. The image sensor consists of 5 basic modules as shown in Fig. 4.13:

1. Pixel array.

2. X–Y-addressing logic.

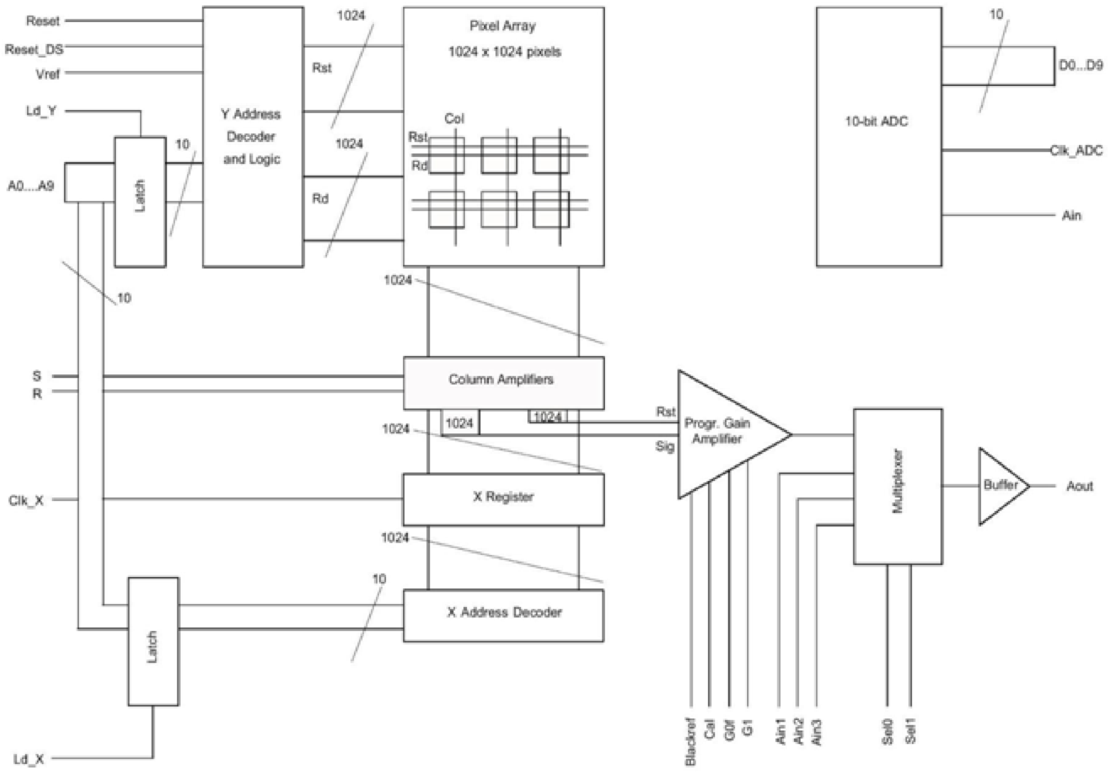


Figure 4.13: Internal architecture of the Star1000 image sensor. It consists of various digital circuits necessary to sequence pixel accesses and to convert the analog output of pixel into digital values.

3. Column amplifier.

4. Output amplifier.

5. Analog to Digital Converter (ADC).

The pixel array is the light-sensitive part of the image sensor. It consists of 1024 × 1024 pixels each of size $15\,\mu\text{m}\times15\,\mu\text{m}$ (Table 4.6). The pixel to be read out is selected using the address value given on the address bus and a signal to the *Ld_X* and *Ld_Y* pins (Fig. 4.13). The $X-$ and $Y-$ addressing logic decodes the location of the pixel from the address bus. The column amplifier samples the output voltage and the reset level of the pixel whose row is selected, and presents these voltage levels to the output amplifier. The output amplifier combines subtraction of pixel signal level from reset level with a programmable gain amplifier. Finally the ADC converts the analog output from the output amplifier to a digital value. It converts the analog value to a 10-bit digital value which can be sampled by the FPGA. The FPGA board is programmed to generate clock signals to control these modules. These clock signals control the image acquisition from the image sensor.

4.4.3.2 Image sensor readout sequence

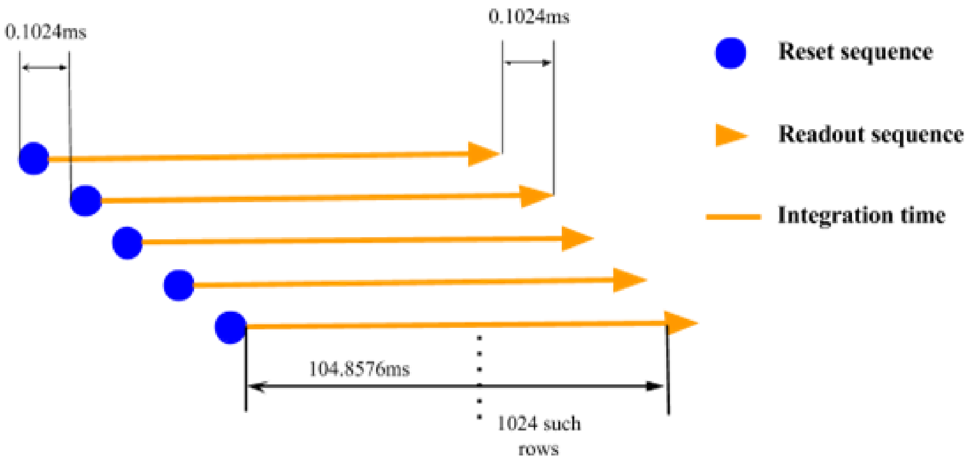


Figure 4.14: Sequence diagram of reset and readout processes

There are two processes to acquire image data from the sensor.

1. Reset

2. Readout

The image acquisition process is implemented into two separate processes: reset and readout. The reset process is done row-wise and the readout process is done pixel-wise. The readout process is done in two parts: row readout and pixel readout. The clock signals for reset process, row readout process, and pixel readout process are described below. A typical reset procedure includes applying proper Y- address (Y- address of the line to be reset) and then giving a reset signal. The integration time of a line is equal to the time between last reset of the row and the time when it is selected for readout. A typical readout procedure includes applying proper Y-address (Y- address of the line to be readout), a sequence of S, R and Reset pulses and then a sequence of X- address and Clk_x and clk_adc to get each pixel digital value presented on the data pins. A basic timing sequence for the readout of the image sensor is shown in Fig. 4.14.

- Row reset sequence: A row in the image sensor can be reset by the following sequence (illustrated in Fig. 4.15):

  1. Place proper $Y-$address on address pins.
  2. Assert *Ld_Y* and latch the address into the internal decoder circuitry.
  3. Pulse the 'Reset' pin so that the internal decoder circuitry resets the $Y-$row in the image sensor.

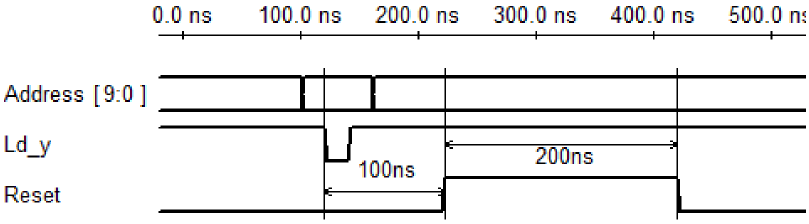


Figure 4.15: Timing diagram for row reset process.

- Row readout sequence: After reset process is done and after the integration time is elapsed, each row must be read. By this process, the outputs of the pixels in the row are connected to an array of column amplifiers. The signals required to achieve this are illustrated in Fig. 4.16:

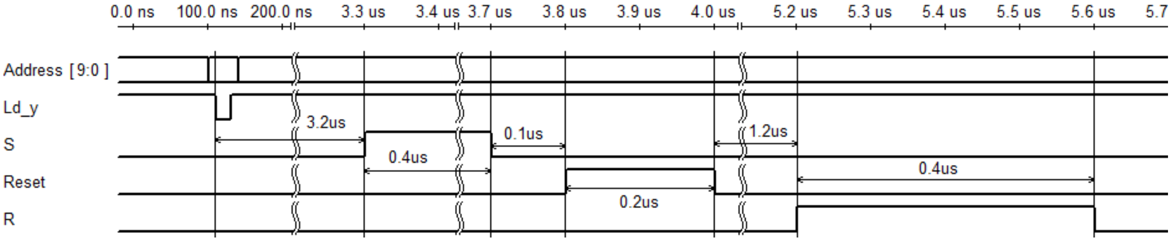


Figure 4.16: Timing diagram for row readout process.

1. Place proper $Y-$address on address bus.
2. Assert Ld_Y and latch the $Y-$address into the internal decoder circuitry.
3. Pulse S signal which will sample values from all column pixels in the row into the column amplifier. This is the signal value after the integration time.
4. Pulse 'Reset' to reset the row in the pixel array.
5. Pulse R signal which will sample the reset values of all column pixels in the row into the column amplifier. Eventually the output amplifier takes the difference between the reset level and the signal level and sends the analog output. Thus, a basic form of correlated double sampling is implemented in the readout process itself.

- 'Cal' pulse to initialize the output amplifier: For every frame there at the first row readout, the 'Cal' signal should be pulsed (illustrated in Fig. 4.17). This gives the black reference (defined by the analog level at 'Blackref' pin) value as output and thus calibrates the complete frame readout. The position of 'Cal' pulse is with respect to the S pulse and can be checked in the datasheet.

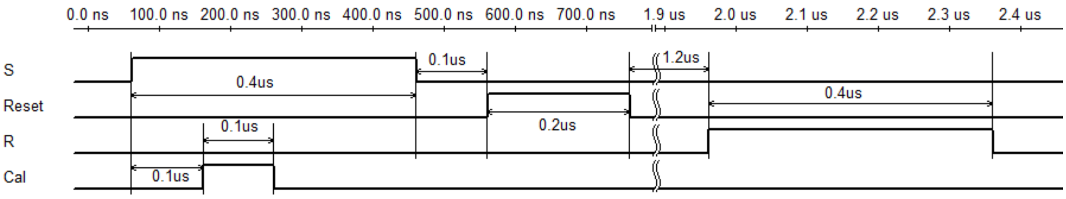


Figure 4.17: Timing diagram for Cal pulse in row readout process

- Pixel Readout/Column readout sequence: This process connects each pixel sequentially to the output programmable gain amplifier and gives reset pulses for the ADC to start conversion. The timing diagram for this process is illustrated in Fig. 4.18.

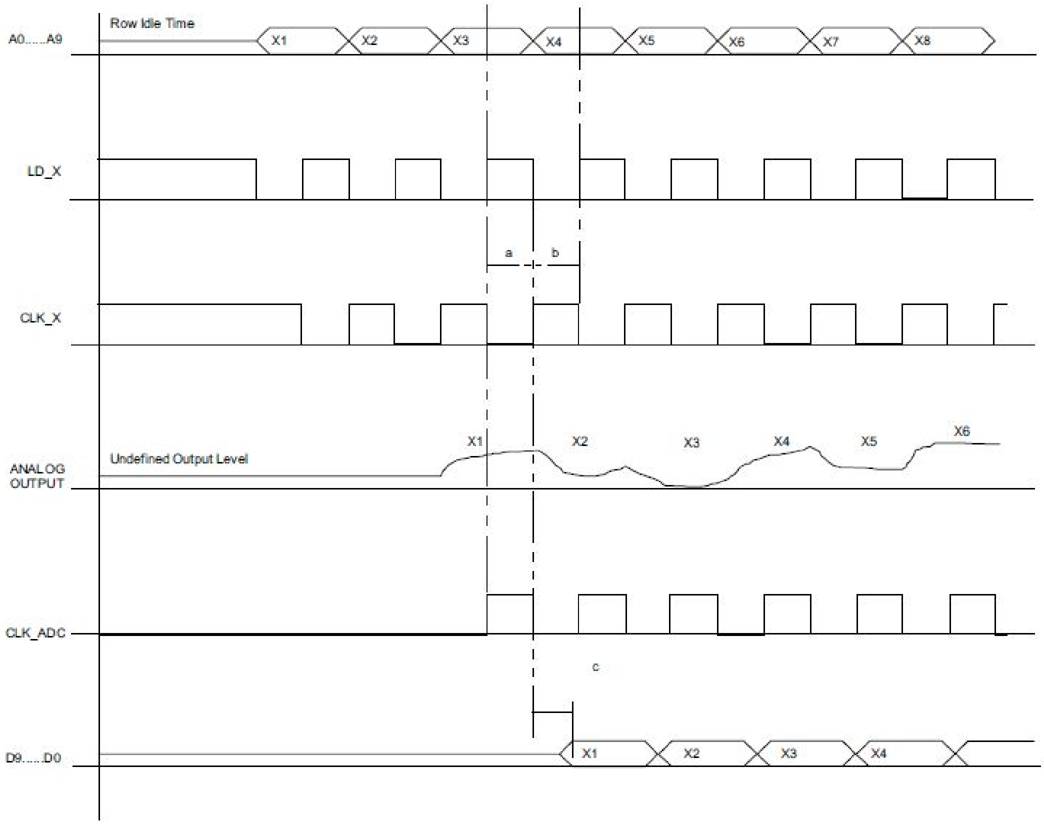


Figure 4.18: Timing diagram for column-wise pixel readout process [27].

4.4.4 FPGA design

The program for the FPGA is written in a hardware description language called Verilog. Modules of Verilog code written to implement a particular task

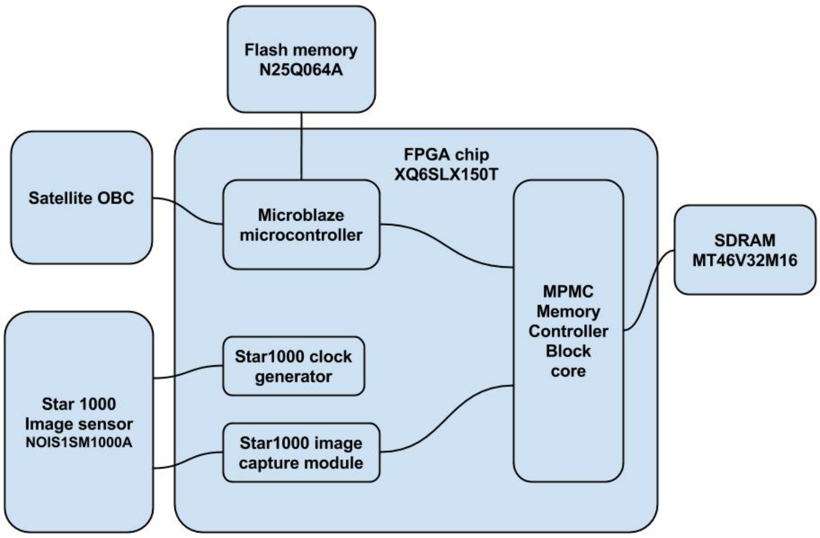


Figure 4.19: Block diagram of hardware logic implemented on FPGA. This consists of the custom IPs developed to interface with the image sensor and integration of these IPs with a multi-port memory controller IP and a microblaze IP to implement the complete design on the FPGA.

are called IP (Intellectual Property). We developed various IPs for interfacing of the image sensor, SD card, SD RAM etc. with the FPGA chip. We also implemented a soft core microprocessor[2], called microblaze, inside the FPGA. The program for microblaze is written in C. The Verilog code is synthesized and a configuration file is generated using the programming toolsuite Xilinx Platform Studio from Xilinx.

A generic block diagram of the logic implemented on the FPGA is shown in Fig. 4.19. We developed 2 main IPs for interacting with the Star 1000 image sensor: Star 1000 clock generator module and Star 1000 image capture module. The image acquisition process as described in the previous subsection 4.4.3.2 is implemented using these IP cores. The clock signals required viz. $clk_x, reset, s, r, Ld_x, Ld_y$ and cal signals are generated by the Star 1000 clock generator module. Where as the sampling of the digital data from the image sensor and dumping into RAM is done by the image capture module. The microblaze microprocessor IP is used to control these modules and implement all the algorithms on the acquired image. It also interacts with the satellite or balloon payload OBC to accept commands and respond to various commands.

[2] http://www.xilinx.com/products/design_resources/proc_central/microblaze_faq.pdf

It is a 32-bit microprocessor with a Reduced Instruction Set Computer (RISC) architecture and the software which is used on the microprocessor can be developed using Software Development Kit which is provided with the Xilinx FPGA development suite.

We use a memory interface block IP called as Multi Port Memory Controller (MPMC) to interface with the external SDRAM (MT46V32M16) provided on the FPGA board. The multi port memory controller allows to share the same memory area between multiple IPs implemented on the FPGA. In our case we share the same SDRAM space with the Star1000 image capture module and the microblaze microcontroller. The pixel data captured by the image capture module and dumped in the SDRAM can be accessed by the microblaze microcontroller and image processing tasks can be implemented on the image data.

The flash memory being non volatile in nature stores the bitstream file for the FPGA program as well as the catalog data. At startup of the system, the microblaze loads the catalog contents from the flash memory into RAM for quick access of catalog data.

The electrical interface with the satellite or balloon payload OBC is implemented on a I^2C standard protocol as well as an industry standard RS485 interface protocol. I^2C protocol is a standard serial protocol introduced by Philips in which a master device can communicate with various other slave devices based on their slave address. It is used in various other sensors. I^2C protocol was selected for flexibility and ease of interface with the OBC. An RS485 interface protocol is a differential signaling protocol, which is widely used in the industry. It allows multi-point communications with each node being assigned an address. A standard packet protocol like ModBus or ProfiBus can be easily implemented on RS485 interfaces to allow effective communication between multiple devices on the bus.

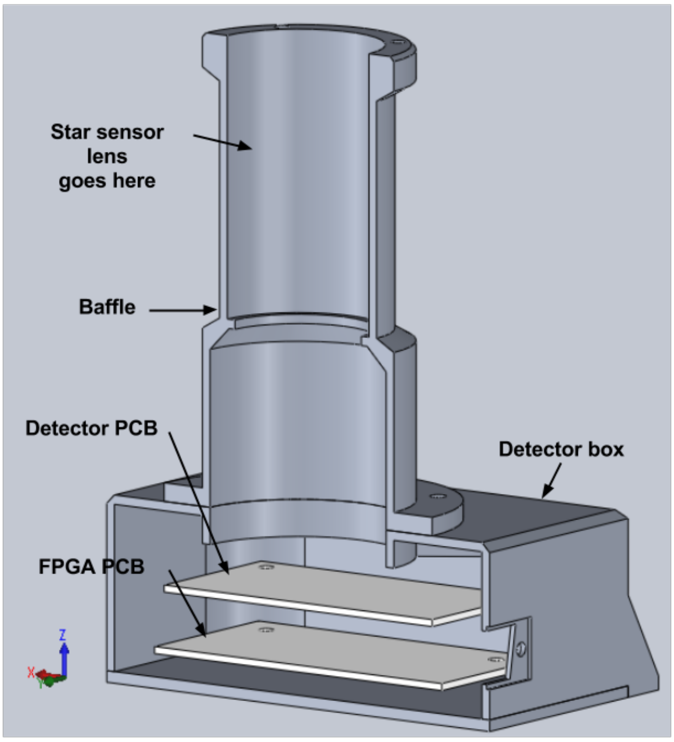


Figure 4.20: Cross sectional view of the CAD model of the *StarSense* assembly structure

4.5 Mechanical design

4.5.1 Housing design and analysis

A structure to assemble the lens and electronics in alignment in a single unit was designed and fabricated. A cross section of the structure and the placement of various components in the assembly is as shown in the Fig. 4.20. The structure consists of 5 parts: primary baffle, detector box, detector box lid, locking ring and spacers to hold the PCBs inside the detector box. The primary baffle holds the lens assembly in place with silicone rubber gaskets. This primary baffle assembles on the detector box which has a hole in the center to allow light from the lens to reach the detector. The detector box holds the detector PCB as well as the FPGA PCB. A lid is used to close the box from back. The detector PCB is mounted in the detector box with adjustable length spacers where as the FPGA PCB is mounted on the lid. The PCBs are connected to each other with a FPC cable. Thus the connection between the detector PCB and FPGA PCB is non-rigid. Therefore, the detector box can be easily opened without disturbing the alignment of the detector PCB. Adjustable length spacers are used to mount the detector PCB inside the detector box to allow precise focusing of the lens to various distances from the camera. Shims are used at the spacer joints for fine adjustment of the focus.

The electrical interface to the sensor is through a 9 pin D type connector on one of the sides of the detector box. The mechanical interface of the structure with the satellite or balloon payload is through 4 M4 nut bolts provided at the corners of the detector box as shown in Fig. 4.21.

The complete structure was designed to weigh as low as 620 gms. A table of the weights of different components of the *StarSense* is as shown in Table 4.9

Component	**Weight (gms)**
Detector box lid	78
Detector box	196
Primary baffle	120
Lens locking ring	10
10mm Spacer	2
5mm Spacer	1
PCB weights (detector + FPGA)	65
Lens assembly	113
Miscellaneous	33
Total	**620**

Table 4.9: Weight breakup table of components of *StarSense*

A satellite undergoes heavy vibrations and static forces during launch. Under these enormous vibrations and launch loads, mechanical systems can disintegrate and stop functioning correctly. Hence, during the design of each of these components, finite element analysis is required to be carried out on the mechanical structure design/CAD model and the structure modified in cases of failures during the analysis. The results of the analysis are useful as reference data points during the vibration testing of the system. The design process of a mechanical system involves an iterative process where designing and analysis are repeated iteratively with modifications in the design in each iteration based on the failures in the analysis. A criteria of keeping natural frequency as high as possible from 100Hz was used in the analysis (Ramakrishnan S et al, 1999).

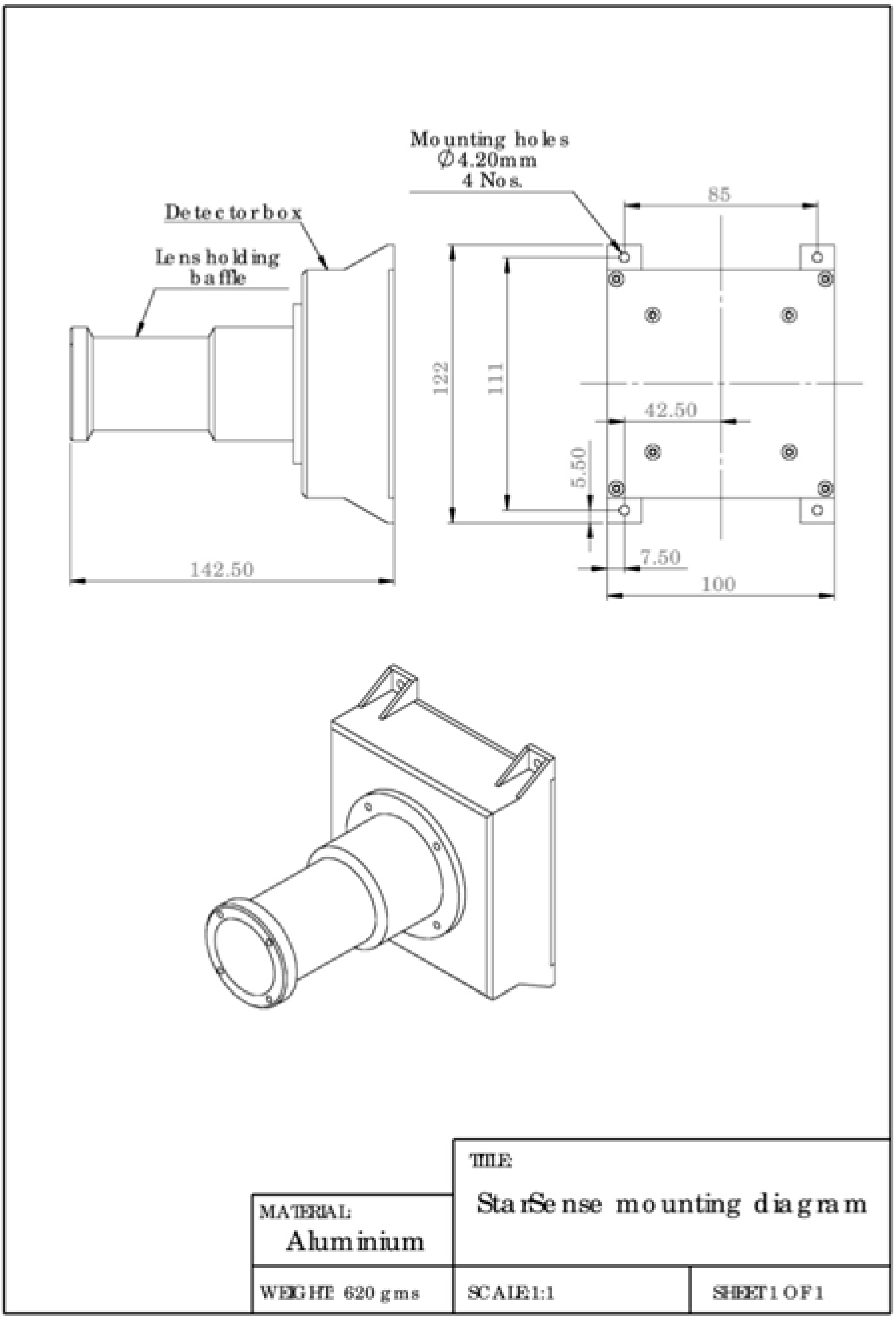


Figure 4.21: Mechanical interface for mounting *StarSense* . The positions of the holes and their dimensions required to mount the sensor to a pointing system or test equipment. This mounting interface was necessary in designing various test fixtures.

Finite Element Analysis is a numerical method for solving engineering problems described by partial differential equations. Complicated geometries are discretized in to finite elements and solved by converting differential or partial differential equations in to algebraic equations which can be easily solved on computers. Modal analysis was carried out to find out the different modes of free vibration and their respective frequencies while the *StarSense* is held clamped at the base. This process consists of 2 steps: meshing and analysis. Meshing a CAD model divides it into smaller elements which can be individually considered as rigid body. These elements are connected to each other at points called nodes. In this case, the meshing was done using 1st order tetrahedral elements all over the structural components. The joints appearing in the structure viz. nut and bolts etc. are modeled using RBE2 Rigid Body Elements. Further boundary conditions were applied on the meshed structure to constrain degrees of freedom of the mounting points of the *StarSense* . Finally the analysis was carried out in MSC Nastran and various modes of vibrations seen in the structure noted down. The results from this preliminary modal analysis for the structure are tabulated in Table 4.10. Note that the natural frequency of the first mode of the structure lies at around 468 Hz which is very much far away from the 100 Hz criterion selected in the first place. This criterion is specified based on the fundamental acoustic vibration of the rocket engine. Higher fundamental frequency of the structure assures that these modes will not be excited due to dynamic coupling between the low frequency dynamics of the launch vehicle and the payload.

Mode	**Frequency (Hz)**	**Observed on**
1	468.4	FPGA PCB
2	495.2	Secondary baffle
3	510.9	Secondary baffle
4	524.1	Detector PCB
5	861.4	FPGA PCB

Table 4.10: Modal analysis of *StarSense* structure

4.5.2 Baffle design and analysis

In operating conditions in space, there are 3 main sources of light that might hinder the desired operation of the sensor: sun, reflected light from moon and earth albedo. If any of them are in the field of view, sensor being very sensitive would get saturated and would not be able to detect light from stars. In cases if the sun's image is being continuously focussed on the image sensor, it might even damage the image sensor due to heating effect caused from sun's light. To reduce this effect to minimum the hot mirror is used which cuts off light in the ultraviolet and infra red wavelength ranges and allows only optical wavelengths to transmit to the detector.

These bright sources of light may cause ghost images even if they are not into the field of view of the optics. These ghost images are generated due to reflection of light from these bright sources at the lens surfaces and walls of the lens assembly. These ghost images add to the noise in the detector and may avoid proper detection of faint stars which are being imaged at the same location on the detector.

To avoid these ghost reflections, a baffle is used. A properly designed baffle prevents stray light from these sources from entering the optical system and contributing to noise. A baffle can be designed with constraints on the angular separation of the bright sources with respect to the optical axis of the system, called as avoidance angle. A narrower source avoidance angle would make the sensor more rugged in terms of operational capability close to a bright source, but typically would increase the length of the system. A wider source avoidance angle would reduce the length of the optical system but would limit the operational field of the sensor.

For *StarSense* , a baffle was designed considering stray light from 2 main bright sources sun and earth albedo (Shahram Mohammadnejad et al (2012), Javad Haghshenas (2014)). We considered a 45 deg sun avoidance angle and 30 deg earth avoidance angle, for this baffle. For a different sun and earth avoidance angle constraints, a different baffle has to be designed. The baffle was designed such that it can be easily attached on top of the *StarSense* assembly. It essentially consists of vanes placed inside a cone at designed locations which prevents light from the bright source reaching the detector directly or after reflections. In this analysis we ensure that no light from the bright source reaches the optical surfaces of the lens without atleast 2 reflections on the baffle cone surfaces. The surface of the baffle with black anodization is assumed to have a reflection coefficient of ~ 0.05. This reduces the light intensity of the bright source to $\sim 0.25\%$ of the original source intensity. The design details for the baffle are shown in Table 4.11.

Vanes	**Position**	**Radius**
1	157.577	75.977
2	111.071	49.127
3	81.001	31.766
4	61.557	20.540
5	61.242	20.358
6	27.018	17.364
7	9.575	15.838

Table 4.11: Schematics of Baffle design

4.6 Conclusion

After having assembled the lens and the image sensor along with the readout circuitry of the image sensor in the *StarSense* structure, the hardware setup for the *StarSense* was complete. Fig. 4.22 shows the assembled sensor. We could capture some real life images with the camera. A Graphical User Interface was developed to control various parameters of the camera like integration time, gain etc. and capture images from the camera and display them on the computer. Fig. 4.23 shows a sample image captured by *StarSense* camera. The camera using a monochrome image sensor would produce a greyscale image. After assembling the constituent components of the *StarSense* hardware in the structure, various calibration tests were conducted on the camera. Determining various camera parameters accurately is crucial for valid operation of a star sensor.

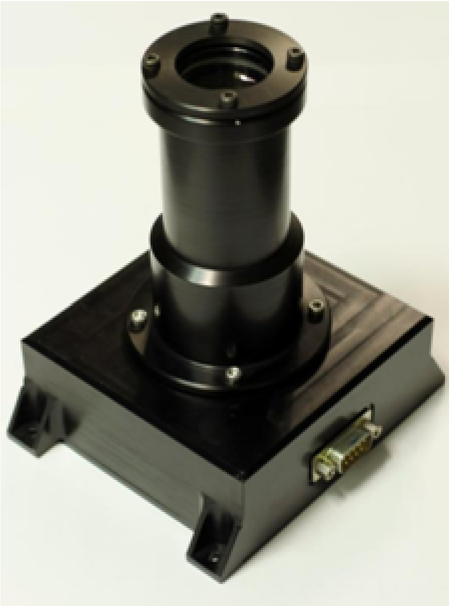


Figure 4.22: An image of the completely assembled *StarSense* .

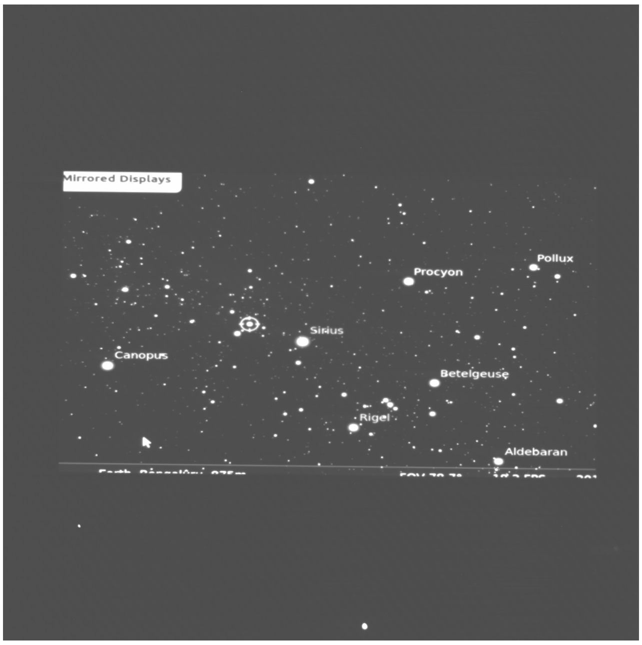


Figure 4.23: A Sample image captured from *StarSense* camera. A view of the software *Stellarium* was projected on a far away screen using an overhead projector. This image shows the capability of *StarSense* to focus on far away objects beyond it's hyper focal distance.

Chapter 5

Calibration of *StarSense*

5.1 Introduction

StarSense is a significantly wide field of view camera which is sensitive to faint stars with up to 6.0 visual magnitude. For the operation of the star sensor, identification of stars is a crucial step. Various identification methods as described previously are dependent on the representation of several geometrical features in star patterns detected in the image. To correctly identify the features in the star patterns by comparing them to corresponding features in star catalogs requires the precise knowledge of the *StarSense* camera parameters. Thus, it is necessary to know the camera calibration parameters like focal length, principal point coordinates, distortion coefficients, sensitivity to star light etc (Hao et al , 2005). We can obtain these values from the optical design of the instrument and various simulations done on it. But, there are multitudes of inaccuracies in the fabrication and assembly of these optical components. This necessitates calibration of the sensor after fabrication and assembly. For wide field of view cameras, the optical aberrations are significantly more than narrow field cameras/ telescopes. Thus, special methods are required to obtain the calibration parameters of wide field cameras (Pengju et al , 2011).

Even after perfect calibration on ground, the placements of lenses, detector and the housing can relatively shift during the launch phase of the space craft. Also, temperature cycling conditions in the orbit of the space craft can cause various components to expand and contract and in turn cause the camera calibration parameters to vary slightly. For high accuracy star sensors, these minor variations are also significantly important to be considered and hence, an on-orbit mechanism to calibrate these cameras is also desired (XIE Junfeng et al , 2009). Although these variations are considered during the design phase in tolerance analysis of the optical design, knowledge of these values in imperative in proper functioning of the algorithms.

This chapter introduces the main primary camera parameters to be considered and measured for camera calibration. It also describes the methods by which these parameters are determined in our case and the accuracies of the calibration values obtained and their validity.

5.2 Detector calibration to quantize detector noises

The detector is an important component of the *StarSense* camera. We have used a CMOS detector in *StarSense* which results into more noisy images as compared with CCD image sensors.

The response of every pixel in the detector is not same for the same intensity of light falling on them. These variations in the pixels are termed as fixed pattern noise. It consists of 2 components dark signal non-uniformity (DSNU) and photo-response non-uniformity (PRNU). DSNU gives a measure of the offset error of pixels across the image sensor plane with respect to the mean value. PRNU gives the sensitivity difference between pixels across the image sensor plane. Together with the vignetting effect of the lens this is called flat fielding effect. In addition to this, each pixel has an average offset error which results due to the biasing errors in the analog electronics and the range setting of analog to digital converter in the image sensor. Broadly, all the errors in the detector can be categorized into the following categories:

- Bias error
- Dark noise
- Flat field noise

To adjust the exposure, we have 2 variables in the detector: 1. Integration time and 2. Gain value. The integration time of the detector is dependent on the clock signals sent to the image sensor for the image acquisition. In the camera readout IP developed for the FPGA, an arrangement is made to send clock signals corresponding to 4 different integration time values (0.1s, 0.2s, 0.5s & 1s). The gain value of the image sensor is dependent on the setting of the programmable gain amplifier inside the image sensor chip. This can be adjusted using 2 pins on the image sensor package. Thus, there are 4 possible gain settings on the image sensor (1×, 2×, 4×, & 8×). The above mentioned detector errors vary with the exposure settings used for capturing images. Hence, calibration tests were carried out to determine the effect of exposure variation on the amount of these errors. A combined metric consisting of the mean value and standard deviation of the pixel intensities was evaluated instead of each error separately. Also the histogram of these images were studied to understand the pixel intensity distribution on the detector.

5.2.1 Detector calibration tests

Various tests were conducted on the detector which constituted of either dark condition on the detector or varying light intensities on the detector. The test procedure and their results are as mentioned below:

5.2.1.1 Dark images and effect of exposure variation

The detector lid was covered and setup arranged in a dark optics lab with temperature maintained at constant value. Images were captured on the computer using a USB connection between the *StarSense* camera and the computer. 10 images were captured at every gain setting and integration time setting possible on the camera. Results of this test are tabulated in Table 5.1 and 5.2. It

Table 5.1: Mean of pixel intensities in dark conditions at different gain and integration time values

	Integration time setting			
Gain setting	0.1s	0.2s	0.5s	1s
1x	77.8507	79.6219	80.2357	81.7732
2x	79.2721	80.0994	80.1178	83.1841
4x	74.1903	74.5992	73.7353	77.8929
8x	66.6194	65.8620	62.6853	69.4833

Table 5.2: Standard deviation of pixel intensities in dark conditions at different gain and integration time values

	Integration time setting			
Gain setting	0.1s	0.2s	0.5s	1s
1x	0.6911	0.8510	1.1952	1.8989
2x	1.3300	1.6374	2.6689	4.2343
4x	3.5905	3.9280	5.7545	8.0435
8x	7.0189	8.3057	11.3447	15.1653

can be noticed that the noise level *i.e.* standard deviation of dark frame noise, goes on increasing with the increasing exposure time as well as gain setting. The mean values remain more or less constant because these are essentially offset values at zero input light flux and that they would not be affected directly by integration time and gain variations. This result is to be taken into consideration while selecting the threshold value for noise in centroiding algorithm (Li, Xuxu, Xinyang Li , 2015). The threshold value used for segmenting the image in the centroiding algorithm is decided based on the standard deviation and mean value of pixel intensities in the dark images. The pixel intensity of 5 times of the standard deviation above the mean value is selected as the threshold value. Thus, pixels brighter than this threshold are to be selected as potential stars for centroid detection.

5.2.1.2 Linearity of pixel response with gain variation at constant integration time

For the subsequent experiments a monochromator was used in conjunction with an integrating sphere. This test is carried out without using the lens

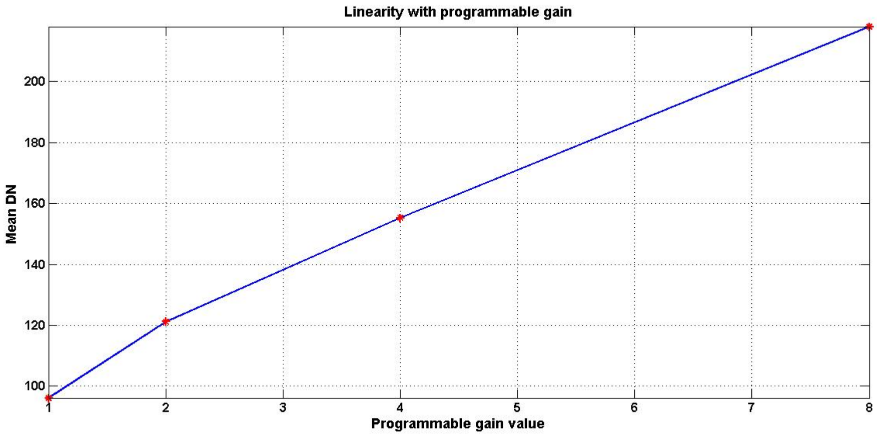


Figure 5.1: Variation of the mean pixel intensities with increasing gain setting

assembled in the *StarSense* assembly. The monochromator was set to emit a wavelength of 590nm where the image sensor response is expected to peak according to the data sheet. The light from the monochromator is fed into an integrating sphere which generates a flat field illumination for the detector. For this experiment, the integration time for the detector was fixed at 0.1s and the gain value was varied over the 4 possible settings (1x, 2x, 4x and 8x). 10 images were captured for each of the gain setting. The mean value of the pixel intensities over the 10 images at a single gain setting was noted down. With the increase in gain, a linear increase in the mean value is observed as can be seen from Fig. 5.1.

5.2.1.3 Linearity of pixel response with integration time at constant gain

Further we evaluated the linearity of the detector with integration time. The exposure being directly proportional to the integration time, the mean signal in ADU should increase linearly with increasing integration time. This test was conducted at a constant gain setting of 1×. Integration time was varied over the 4 possible values 0.1s, 0.2s, 0.5s and 1s. 10 images were captured for each of the integration times. Finally a mean value of each of the 10 images was noted and plotted. The plot showing the linearity of variation over integration time can be seen in Fig. 5.2.

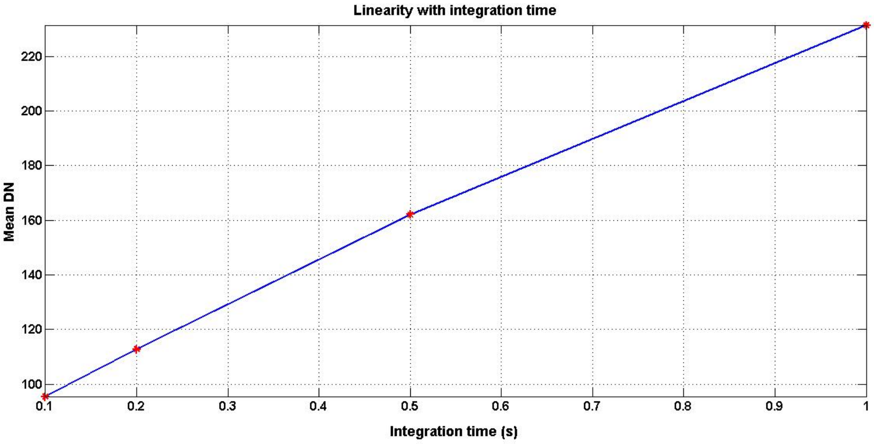


Figure 5.2: Variation of the mean pixel intensities with increasing integration time

5.2.1.4 Signal mean variance curve and CMOS gain

The gain of the CMOS detector in units of e^{-}/ADU is an important quantity to be determined for an image sensor. This defines the number of electrons required to output 1 LSB digital number outside of the ADC. This gain value of the CCD can be obtained by a practical statistical experiment using the mean and variance values of the detector ADU response over varying source brightness at a constant exposure setting. We conducted a gain calibration test for the CMOS imager. This was done using an integrating sphere with a variable source. The integrating sphere was used to achieve a flat field illumination over the detector and the variation in the experiment was achieved by adjusting the slit width of the source feeding the integration sphere, thus in turn changing the intensity of the input light. The gain (e^{-}/ADU) can be estimated by using the mean variance characteristics of the images at different illuminations. The experiment was repeated over 4 different digital gain values. The estimation of the gain works on the principle as seen in eqs. 5.3. The underlying principles for CCD/CMOS image sensor based astronomy used in these equations are as described in the course PHY217 (2014).

$$S_e = g.S_c, N_e = g.N_c\,, \tag{5.1}$$

$$N_e^2 = R_e^2 + \sigma_e^2 + \sigma_{0,e}^2 = R_e^2 + S_e + \sigma_{0,e}^2\,, \tag{5.2}$$

$$N_c^2 = \frac{1}{g}.S_c + (R_c^2 + \sigma_{0,c}^2) \tag{5.3}$$

Here S_e, N_e represent signal and noise respectively in electron units, g is the gain value which is to be determined, and S_c, N_c represent signal and noise respectively in ADU (or counts) units. We can express total noise as sum of squares of readout noise (R_e^2), photon noise (σ_e^2) and other flat field noise ($\sigma_{0,e}^2$). But knowing that the arrival of photons from a star in a collecting area is a Poisson process and for such a process, signal value is the same as the noise variance ($S_e = \sigma_e^2$). Replacing these values in the equation gives a relation between the noise variance and signal mean value. Considering that the readout noise and flat fielding noise remains constant for a given exposure setting, the equation denotes a straight line with a slope of $\frac{1}{g}$. Thus, if we plot the mean and variances over varying source brightness on a single graph, the slope of the line fitted in the observations can be used to determine the gain of the detector in e^-/ADU units.

We determined the mean and variance of signal in images captured at fixed illumination from the integrating sphere and obtained their variation by varying the source slit width. A straight line equation was fitted in the observations to get the gain value of the CMOS pixels in the image sensor (Fig. 5.3). After obtaining the CMOS gain values for each of the gain settings available in the image sensor, we tabulated the expected signal over the complete PSF in ADUs for star magnitudes from 1.0 to 6.0 as can be seen from Table 5.3. The table shows only gain settings that are useful for actual observations *i.e.* Gain 2× and Gain 4×. Gain setting of 1× makes the images underexposed and 8× contains a lot of noise thus reducing the effective signal to noise ratio.

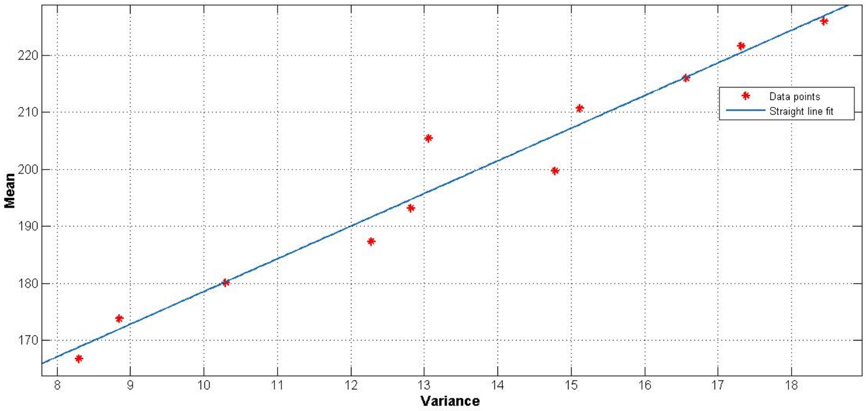


Figure 5.3: Mean variance curve of the detector over varying illumination from integrating sphere

Table 5.3: Expected signal in ADUs from different limiting magnitude stars

Star magnitude	Gain - 2x CMOS gain = 7.356	Gain - 4x CMOS gain = 5.193
1	70312.16	99598.74
2	27991.78	39650.97
3	11143.73	15785.34
4	4436.40	6284.26
5	1766.16	2501.81
6	703.12	995.99

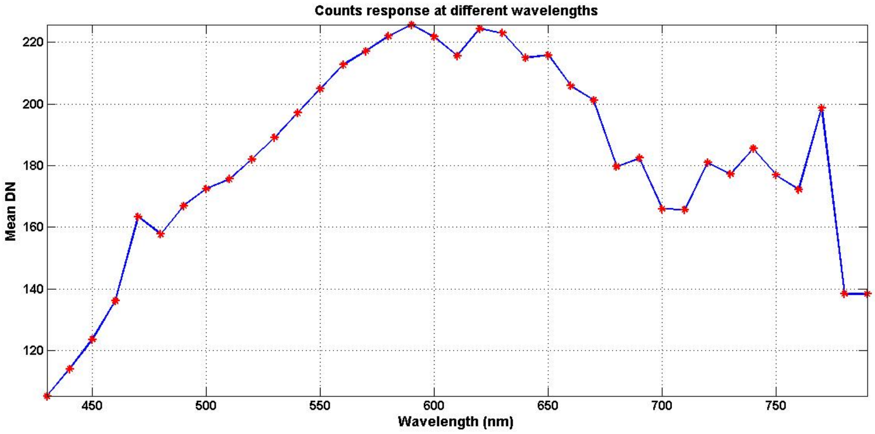


Figure 5.4: Spectral response of the *StarSense* camera as observed in digital ADUs from the detector

5.2.1.5 Spectral response of the detector

Although, the detector is not being used for carrying out spectral observations of star, an experiment was conducted to get an idea of the spectral response of the detector. The effective response visible from the graph is a combined effect of the monochromator response/ source response and the detector response. The graph is shown in Fig 5.4. The peak response occurs at $\sim 590nm$.

5.2.2 Standard detector calibration process

For on line processing of images, it is necessary to remove the above mentioned noise from the captured images. The single pixel noise (salt and pepper noise) is distinguishable from actual signal based on the spread over multiple pixels. To correct the intensity variation of pixels coinciding with the hot pixels, a standard calibration process was devised. This process uses a standard calibration dark image frame which included the pixel noise variations. A process similar to flat fielding corrections was implemented on each of the images to correct for these standard calibration images. This process of calibration was implemented in hard coded IP module in the FPGA which corrected the inputs in realtime during image capture. This saved precious processor time. This was used as a standard raw image processing step for all the images captured by *StarSense* . This standard correction methodology reduced the possibil-

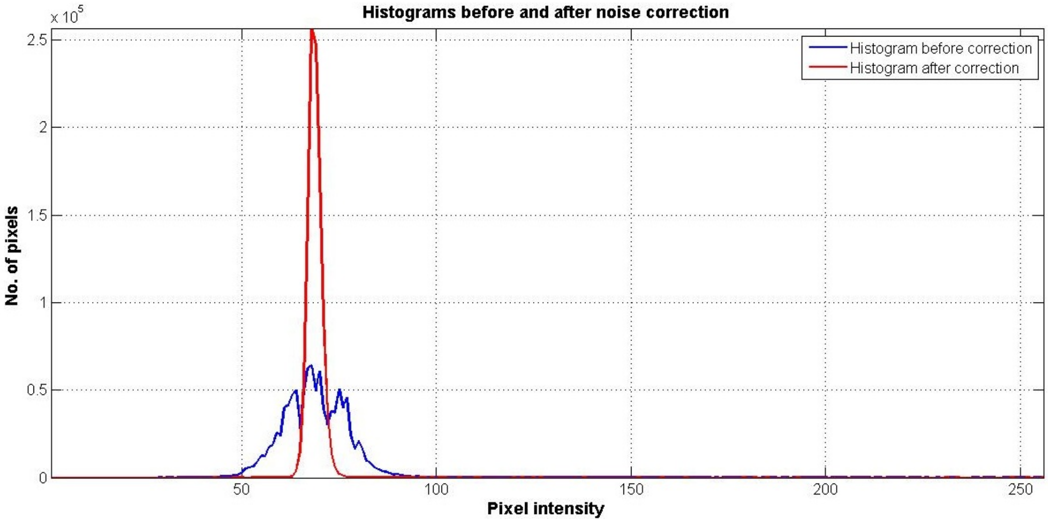


Figure 5.5: Graph showing the effect of calibration using standard calibration frames on the histograms of the dark images. The histogram width drastically reduces after calibrating, thus allowing a higher dynamic range and a sensitivity to fainter stars.

ity of false detection of stars due to salt and pepper noise and also improved the histogram of the intensity distribution in a raw image. A sample graph showing histograms of the images with and without this calibration correction is illustrated in 5.5. The standard calibration frames used varied according to the exposure setting of the camera. The graph shown in the image was obtained for an image captured at integration time of 0.2 seconds and gain of 8x. The main advantage of using such a calibration technique was the clear reduction in the noise spread of pixels in the intensity histogram. This allowed a wider dynamic range to be used for captured images as well as the threshold setting required for star detection was reduced which allowed detection of fainter stars in the image.

5.3 Photon noise characteristics

We know that the process of collecting light from star using a camera is a Poisson process and that the Poisson statistics apply to it. For a given star, the measured signal/counts in multiple exposures of the same star in same environmental conditions will be a Poisson function and will have a mean value corresponding to the brightness of the star and a standard deviation of intensity corresponding to the photon noise. We tried to simulate this and verify it by measuring the signal and noise involved in the process. A laboratory setup was established which simulated a star at infinite distance. An LED light source illuminating a $100\mu m$ fiber was setup on an optical table. The fiber aperture served as a pin hole. The other end of the fiber optic cable was mounted with a rigid fiber holder at the focal plane of a standard lens with a focal length of 50mm. This setup worked as a basic collimator source. The divergence angle of the collimator source because of the pin hole finite dimension was calculated to be smaller than the angular resolution of the *StarSense* camera and thus the pin hole worked as a point source for the camera. The output plane wavefront from the collimator was fed to the *StarSense* camera which imaged it on it's focal plane and a PSF was obtained. The camera exposure was set to a gain setting of 4x and integration time of 0.2s considering the brightness adjustment of the LED light source. The brightness of the LED source was varied over a few steps to simulate different brightness stars. This was done by varying the duty cycle of the pulse width modulated (PWM) signal powering the LED. A circuit was made to obtain constant PWM for a given potentiometer setting. Thus by varying the potentiometer, the brightness of the LED could be varied. 100 images were captured at each LED source setting. The total signal in a 7x7 pixel square area around the centroid was obtained and tabulated for every image. The variation in this signal gave an estimate of photon noise in the imaging process. The results are as tabulated Table 5.4. It can be seen from these results that with higher signal value, the photon noise measured also increases and to be precise suggests the square root relation between signal and the noise.

5.4 Centroiding error

Another important factor to be evaluated is the centroiding error in practice. As described in chapter 3, the centroiding error depends on the signal to noise ratio and reduces with increasing signal to noise ratio. This had to be veri-

Table 5.4: Signal mean value vs standard deviation due to photon noise from source

Signal mean value	Signal standard deviation value
1284.97	41.1520
2789.25	43.7981
3901.95	78.0127
5564.88	85.0753
11082.62	97.7767

Table 5.5: Centroiding error measured by the camera as a function of the signal

Signal mean val	Centroid location	Centroid error(pixels)
1284.97	(497.4618,542.5131)	0.0847
2789.25	(497.6571,542.8176)	0.2268
3901.95	(497.7963,543.0430)	0.0275
5564.88	(497.9588,543.4235)	0.0425
11082.62	(498.2226,543.4201)	0.0494

fied with the real imaging system. The images captured in the previous setup to understand photon noise could also be used for measuring centroiding error and its variation with different source brightness. This was because, the components were setup on a rigid optical table and the relative shift between the collimator and the *StarSense* was negligible. Hence, for practical purposes the star source was at a fixed location on the image plane. The shift in the calculated centroid was primarily due to the noise in the image sensor and the algorithm efficiency. The centroid location and it's shift obtained during the image acquisition process of these 100 images is as tabulated in Table 5.5.

5.5 Star magnitude calibration using standard test setup

After having determined the noise parameters in the detector, it is necessary to obtain a mathematical relation between the star magnitudes and the signal generated in the detector from the corresponding star. This would enable to estimate the signal to noise ratio at different magnitudes and then compare it with the design value for the sensor.

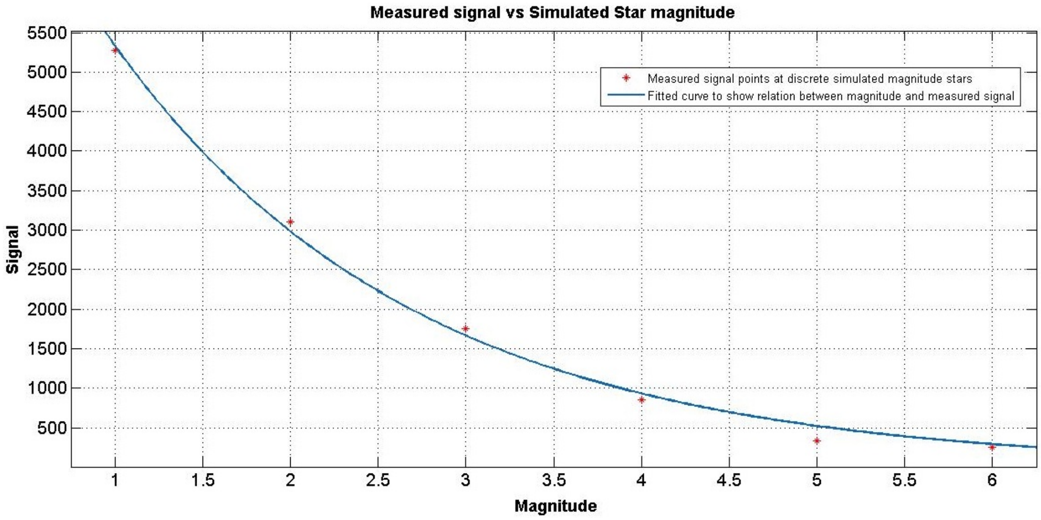


Figure 5.6: Signal vs. Magnitude curve for magnitude calibration done at HASS

5.5.1 Signal from simulated stars of various magnitudes

To evaluate the variation of the signal measured by *StarSense* for different magnitude stars an experiment was conducted with a simulated star source in the lab. This was implemented at the star lab at LEOS, with a test setup called high accuracy star simulator. This test setup consisted of a collimating telescope which collimated the light from a point source object and generated a plane wavefront which simulated the light coming from stars. The intensity of the source was adjusted using neutral density filters with the desired attenuation, to match with that corresponding to a real light source in the sky .

Our experiment involved capturing images of the simulated star at different exposures and tabulating the signal values to obtain the transfer curve between measured signal and actual magnitude. These images were captured at an exposure setting of 8x gain and 0.5s integration time. The measured signal was calculated by taking the sum of pixel intensities around the detected centroid. A square of 7x7 pixels was used to obtain this sum. The 7 pixel criterion was obtained from the measurement of PSF spread as described in Chapter 4

5.5.2 Curve fitting and curve parameters

To obtain the mathematical relation between the actual magnitude and detected signal in the camera, an exponential curve was fitted in the signal vs magnitude points obtained from the simulated star experiment. Exponential curve was selected based on the definition of visual magnitude units as shown in eq. 5.4. Hence an equation of the form as shown in eq. 5.5 was fitted in the plotted data. The coefficients a, b obtained from the curve fitting are as shown in eq. 5.6. The curve fitted using this mathematical expression in the scatter plot of observations is shown in Fig. 5.6. This mathematical expression provides a way to estimate the brightness of the star imaged using the signal obtained from a 7x7 window around the centroid of the star in the image. These experiments were conducted in a lab using controlled sources and setup. An experiment under the open sky was necessary to validate the calibration.

$$m_1 - m_0 = -2.5 \times log(\frac{S_1}{S_2}) \tag{5.4}$$

$$\begin{aligned} f(x) &= a.exp(b.x) && (5.5) \\ &= 9548 \times exp(-0.5821.x) && (5.6) \end{aligned}$$

5.5.3 Signal from real sky image and star magnitudes

Real sky conditions vary from the lab conditions in the various aspects like haze in atmosphere, clouds and moisture in the atmosphere causing attenuation of light from stars, sky background light from moon, air mass and zenith distance etc. However it is still interesting to calibrate the sensor under real sky conditions to obtain a calibration curve. Hence, we carried out a magnitude calibration test for the sensor at an astronomical observatory situated at Kavalur. Images were captured by the *StarSense* pointed at zenith as the stars transited the zenith field. From the timestamps of the images and patterns

Table 5.6: Magnitude of the stars detected and their corresponding measured signals

Star magnitude	Measured signal
3.63	247
5.4	81
5.4	79
1.53	1437
4.52	144
3.98	166
5.78	52
4.82	143

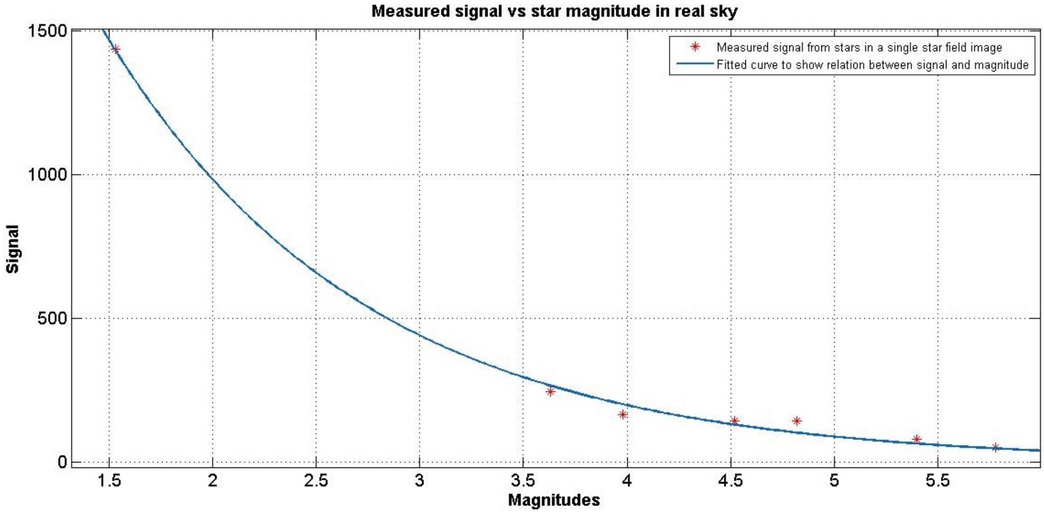


Figure 5.7: Curve fitted in the measured signal readings from images of different magnitude stars in the sky captured using *StarSense* .

of stars in the image, the stars detected in the image were identified using a planetarium simulation software. The *StarSense* was pointed at zenith so that the effect of airmass would be minimal and the apparent magnitude of the star would remain unchanged. The signal obtained from these stars was calculated and is tabulated in Table 5.6 and a curve fitted through these readings is as shown in Fig. 5.7. The signal was measured using the total counts in a 7x7 pixel area around the detected centroids. A 7 pixel window was taken considering the PSF spread as practically measured and described earlier.

5.5.4 Comparison of curve fitting parameters

The measured signal value was plotted against magnitude of the stars obtained by matching them in a planetarium software called stellarium. An exponential curve like the previous exercise for simulated stars, was fitted in the readings. The equation of the curve fit was obtained as given in eq. 5.8. These images were captured at an exposure setting of 4x gain and 0.5 s integration time. Considering the ratio between gains in the two cases laboratory and real sky, the obtained curve fit equation matches very well with the laboratory experiment. Thus it was concluded that the curve fitted equation is the desired equation to estimate the signal for a certain magnitude star.

$$
\begin{aligned}
f(x) &= a.exp(b.x) \qquad (5.7)\\
&= 4895 \times exp(-0.8023.x) \qquad (5.8)
\end{aligned}
$$

5.6 Pin hole camera model

As described in chapter 3, 3 different coordinate systems are used in *StarSense* . The sensor body coordinate system or the camera coordinate system can be modeled using a generic pin hole camera model. Essentially, this model maps the 3 dimensional world around the camera on the 2 dimensional image plane of the image sensor. Thus, this modeling essentially develops a relationship between the image plane coordinate system and the sensor body coordinate system or the camera coordinate system.

The pin hole camera model can be represented as a 4 × 3 matrix which is called the camera matrix. This camera matrix is computed based on intrinsic and extrinsic parameters of the camera. The extrinsic parameters represent the position of the camera with respect to the surroundings where as the intrinsic parameters refer to the focal length and optical center of the camera. In our case, the camera being used for imaging stars, the extrinsic parameters are not

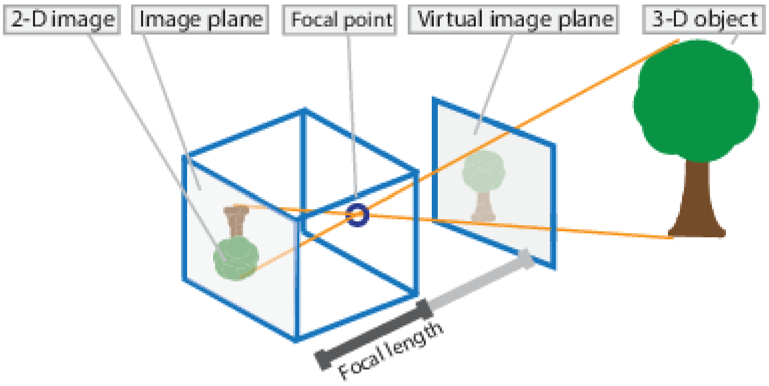


Figure 5.8: Visual definition of a pin hole camera model

used because the distances to stars can be suitably assumed to be infinite with respect to the camera physical dimensions. In addition to the intrinsic camera parameters, the distortion effect introduced by the lens plays an important role in the geometry of the image acquired, i.e. the patterns of stars visible in the image. This distortion effect being a systematic irregularity in the camera can be modeled and safely corrected for to obtain the true patterns in the stars. We describe about the distortion modeling also in the subsequent sections.

As seen from Fig. 5.8, the pin hole camera model is a model of camera where the image is formed using a small aperture and without a lens. The light rays from the surrounding sources pass through the small aperture and form an inverted image on the 2 dimensional image plane behind the pin hole aperture. The distance between the aperture and the 2 dimensional image plane corresponds to the focal length f of the lens. We think of an imaginary plane in front of the aperture at the same distance f on which an upright image of the scene will be formed. The image sensor has the pixel numbering on the x and y axis in an inverse fashion as can be seen from Fig. 5.9, which essentially inverts the image when the pixels are addressed in the regular format. Thus, the data obtained from the image sensor implements the image inversion in the readout process and thus depicts the scene as seen from the imaginary focal plane in front of the aperture.

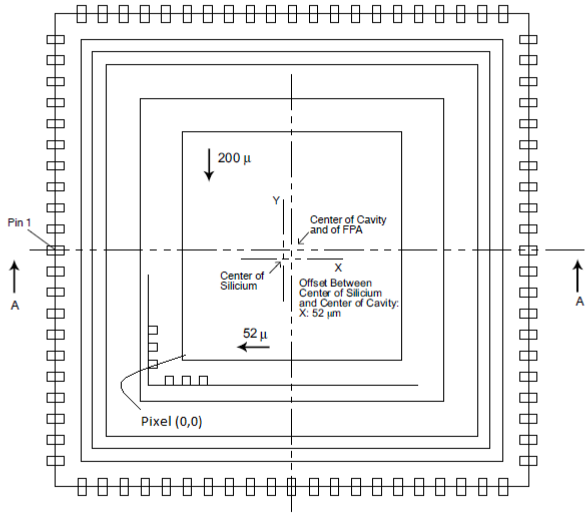


Figure 5.9: Location of pixel (0,0) in the image sensor array. The center of the active area is slightly offset from the center of the image sensor IC. This offset has to be considered while designing the PCB layout of the image sensor.

5.7 Camera model parameters

A camera essentially projects points from the 3 dimensional world coordinates on a 2 dimensional image plane. This projection is unique in the sense that each point on the 2 dimensional plane can be mapped to a unique unit vector in the camera coordinate system. The mapping done by the camera from 3 dimensional world coordinates to 2 dimensional plane is called perspective projection. For a captured star field image, we already have the coordinates of the stars in 2 dimensional plane. Thus, we are interested in an opposite process which determines the unit vectors in the camera coordinate system for the imaged points. This requires precise knowledge of the physical parameters of the camera.

5.7.1 Intrinsic parameters

The unit vector calculation is dependent on various physical properties of the camera like focal length, principal point, pixel pitch etc. The process of calculating the unit vectors from the 2 dimensional coordinates can be mathematically represented as shown in Equation 3.3 in chapter 3. They include all the above mentioned parameters of the camera. To know these parameters for a camera, a calibration process has to be undertaken. These parameters of the camera are called intrinsic parameters.

5.7.2 Extrinsic parameters

The parameters which define the position and orientation of the camera with respect to the surrounding scenes are called extrinsic parameters. In our case we do not need to consider extrinsic parameters because we use the camera always in an infinite conjugate mode i.e. we always capture images of stars which are almost at infinite distance from the camera with respect to the camera focal distance and other dimensions.

5.7.3 Distortion parameters

After having determined the first order geometric effects of the camera, its aberration effects are to be taken into consideration to improve the accuracy of calculation. The most affecting aberration effect in an instrument for angle measurement is the distortion effect. The distortion error introduced by the lens is a systematic error and can be corrected in software. After the centroids are calculated, we implement the distortion correction process to correct for the lens distortion. It essentially is the process to calculate the undistorted centroid coordinates using Eqs. (5.9) and (5.10). The lens being designed in-house, a theoretical distortion model ($[K_1 K_2 P_1 P_2]$) is available from ZEMAX simulation results. A more accurate practical distortion model is obtained using a MATLAB toolbox. The calculated centroid positions are corrected to ideal positions using this distortion model equation. We use a combination of radial and tangential distortion model to correct for the distortions.

$$x_u = x\left(1 + K_1 r^2 + K_2 r^4\right) + P_2\left(r^2 + 2x^2\right) + 2P_1\left(xy\right) \tag{5.9}$$

$$y_u = y\left(1 + K_1 r^2 + K_2 r^4\right) + P_1\left(r^2 + 2y^2\right) + 2P_2\left(xy\right) \tag{5.10}$$

The number of coefficients to be used for correction of centroid coordinate location depends on the residual error after correction. With more number of coefficients in the distortion correction equation, the residual error decreases drastically. On the other hand, more the number of coefficients used for correction, the task becomes computationally more intensive and hence consumes more time limiting the operational rate of the sensor. Hence, an optimization between the number of parameters to be considered and calculation time is required for the application software. This was decided based on the results of a distortion simulation from ZEMAX which is described further in this chapter.

5.8 Methods to determine camera model parameters

One of the advantages of having designed the lens in-house was that the above mentioned parameters of the lens system which are crucial in all calculations were known from the design. With tolerance limits provided on fabrication and measurement capabilities, a probable range for the camera parameters were obtained.

In spite of this practical measurements of all the camera parameters is essential. This is primarily because, while designing the lens, its housing and other aspects like mounting mechanism etc. were not considered. A calibration process for a completely assembled sensor is crucial to eliminate all the accumulated errors and aberrations in the assembled camera.

We used various methods to determine the camera calibration parameters through simulations and through practical experiments. These methods are described further in this chapter. The main parameters considered are focal length, principal point and distortion coefficients. The values of these parameters computed using various methods are finally compared with experimentally obtained values and design values.

The image sensor being an off-the-shelf component, various parameters related to the image sensor were obtained from the datasheet along with some assumptions due to their negligible effects. Pixel pitch and the resolution was obtained from the image sensor data sheet. The variation in the pixel pitch across the image sensor and that the pixels are square in shape rather than rectangular or rhomboidal thus eliminating pixel skew parameter from the list are some of the assumptions made about the image sensor.

5.8.1 Camera parameters from ZEMAX simulation

The lens system used in *StarSense* was designed in-house. It is a multi-element Tessar lens system designed to minimize different aberrations as much as possible up to the sensor accuracy requirements. During the design process various simulations were done using the software ZEMAX primarily to determine the imaging performance of the system viz. point spread function, encircled energy, field curvature and distortion and other primary aberrations. The camera parameters required for *StarSense* operation can also be computed from the software. Table 5.7 shows all the values computed in the software. The principal point of the camera cannot be obtained in ZEMAX. This is because the real location of the image sensor at the focal plane would require accurate information about mounting of the image sensor PCB at the focal plane. This results in no knowledge of the optical axis of the lens system with respect to the detector plane. Thus we can only calculate focal length and distortion coefficients for the lens using ZEMAX.

While designing the lens system, a required value of focal length was first obtained as described in Chapter 3. This required a focal length constraint while designing the merit function in ZEMAX for the lens design. The weightage for the constraint defines how accurately the optimization engine would try to achieve the constraint. Giving a higher weightage, makes sure that the optimization engine achieves this constraint more closely. Thus, the required 80mm focal length is met. The tolerance analysis conducted on the lens was useful to estimate the variation in focal length due to manufacturing tolerances. The corresponding variation in the field of view was also estimated which was negligible.

Table 5.7: Intrinsic parameters of camera as obtained from ZEMAX simulations

Parameter	Value
Focal length	80 mm
Focal length error due to manufacturing tolerances	0.3mm
Corresponding error in field of view	$2''$
Distortion coefficients	k_1=-0.0902

The distortion coefficients were also calculated from outputs of ZEMAX. A grid of field points in the x and y directions of the image plane are used in ZEMAX for distortion modelling. The table lists paraxial (x, y) coordinates and real (x, y) coordinates. The paraxial (x, y) coordinates define the ideal field point positions where as the real (x, y) coordinates define the position of the points considering distortion of the lens. The relation between the paraxial and real coordinate points can be defined from eq. 5.9 and eq. 5.10. An optimization was used to determine the distortion coefficients k_1, k_2 and so on. The analysis showed that the maximum reduction in error was obtained by introduction of a single coefficient k_1 in the correction step. Further inclusion of higher coefficients only marginally improved the distortion error. On the other hand, using higher order coefficients made the distortion correction in real-time more computation intensive. Thus, using only the first coefficient was sufficient for distortion correction. This coefficient is noted in the Table 5.7. Figure 5.10 shows the effect of distortion correction using single coefficient. The left most figure shows the grid of points at the focal plane. The middle figure shows the distance of the shifted grid point due to distortion, the blue curve shows before correction and red curve shows after correction. A zoomed in view of a single grid point is shown in the right most figure. The maximum residual error is $\sim 1.3''$ which is well below the required sensor accuracy as well as pixel scale.

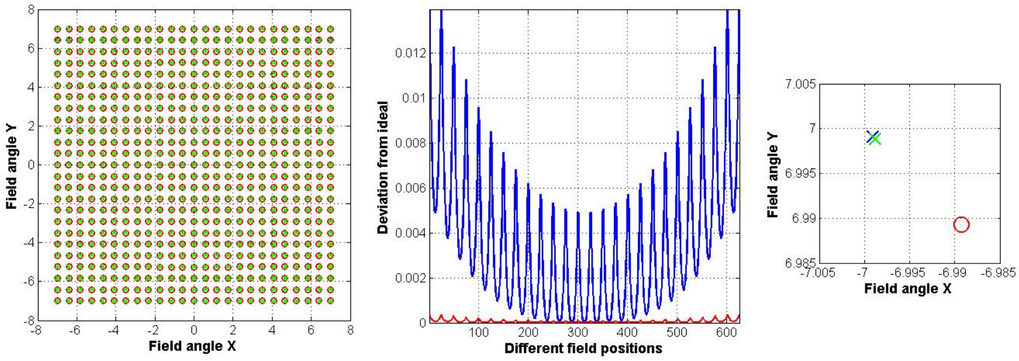


Figure 5.10: From left to right, 1. Full field distortion grid as obtained from ZEMAX, blue points are ideal/paraxial coordinates, red points are distorted coordinates and green points are corrected coordinates using 1 coefficient, 2. Field position error magnitude for different field positions, blue curve shows error for distorted points, red curve shows error for corrected points using 1 coefficient, 3. Zoomed view of maximum distorted point at the edge of the field

5.8.2 Camera parameters from experimental results and MATLAB toolbox

MATLAB software has a lot of different applications which are built for simplifying user tasks. There is a camera calibration toolbox application included in it's suite of applicatons too which is very useful in calibrating camera parameters. Fetic, Azra, Davor et al (2012) describes the use of MATLAB camera calibration toolbox to determine a camera parameters using images captured by the camera under test. Images of checkerboard captured using the camera under test are passed as input to the camera calibration toolbox. The toolbox identifies and calculates locations of the central points in the pattern. Physical parameters of the pattern are also required to be input to the application. From these user provided information, it calculates the intrinsic and extrinsic camera parameter matrix.

The camera is an infinite conjugate system and hence, the object under consideration for calibration has to be put at almost infinite distance with respect to the camera. Hyperfocal distance for a camera is defined as the minimum distance beyond which everything can be considered as infinite distance with respect to the focal length of the camera. This can be obtained from the size of circle of confusion (which in our case can also be assumed to be the point spread function) of the lens system on the detector pixels. For our camera this is calculated as $\sim 40m$. Thus, we need to place the regular pattern at least at a distance of $\sim 40m$. In the current case, we need to input multiple images of a checkerboard captured by the camera under test. For each block of the checkerboard chart to project at least about 20 pixels on the detector plane, and considering that it is placed at a distance of 40 meters from the camera, the size of each block of the checkerboard must be at least $\sim 140mm$. Thus, a checkerboard was printed on an A0 page size to be used as the calibration target for the MATLAB camera calibration toolbox.

This checkerboard was fixed at distance of $\sim 40m$ from the camera in an open environment. Multiple images of this checkerboard were captured using *StarSense* camera. These images are input to the camera calibration toolbox.

5.8.2.1 Camera calibration toolbox operation

The first step in the toolbox pipeline is to identify the checkerboard pattern and detect centroid pixel coordinates of each checkerbox corners. The function detectCheckerboardPoints in the toolbox identifies these corner points of the squares in the checkerboard and assigns index coordinates to each of them in a sequence. The origin (0,0) corresponds to the lower-right corner of the top-left square in the checkerboard. After having calculated the centroid coordinates for each corner, the toolbox generates an ideal grid for the same dimension checkerboards with same position of the camera in the world coordinate system. This is done using the generateCheckerboardPoints function in the toolbox. The error between the imaged positions and the ideal positions is calculated. By fitting values to the various parameters of the camera, the re-projection error is calculated.The re-projection error is the error between the imaged grid coordinates and the ideal grid coordinates after considering effects of the camera parameters. An optimization is run to minimize this re-projection error and finally the intrinsic and extrinsic parameters of the

Table 5.8: Intrinsic parameters of camera as calculated from MATLAB camera calibration toolbox

Parameter	Value
Focal length	80.24mm
Principal point	(540,624)
Distortion coefficients	-0.1542
Skew	0

camera are obtained.

5.8.2.2 Camera calibration toolbox results

The values of the essential camera parameters from the use of camera calibration toolbox are as shown in Table 5.8. The focal length obtained from the camera calibration toolbox corroborates with the design value and tolerance as estimated in ZEMAX. The principal point location is an additional information which is estimated using the camera calibration toolbox which could not be simulated in ZEMAX. Distortion coefficient considering only one coefficient for correction of distorted images, aligns with the estimated value in ZEMAX. The image sensor having square shape pixels and perpendicular in direction, the skew parameter obtained is zero. Thus the values are comparable to the values obtained from ZEMAX during the design stage. Thus, the camera calibration toolbox gives the consolidated effect of all the errors in the camera imaging system.

5.8.3 Camera calibration tests using High Accuracy Star Simulator

We also conducted practical tests at a standard test facility at LEOS, ISRO, to calibrate the camera very accurately and know the required parameters with high level of accuracy. These tests also helped in determining the relative accuracy with which other methods could be used to calibrate the camera.

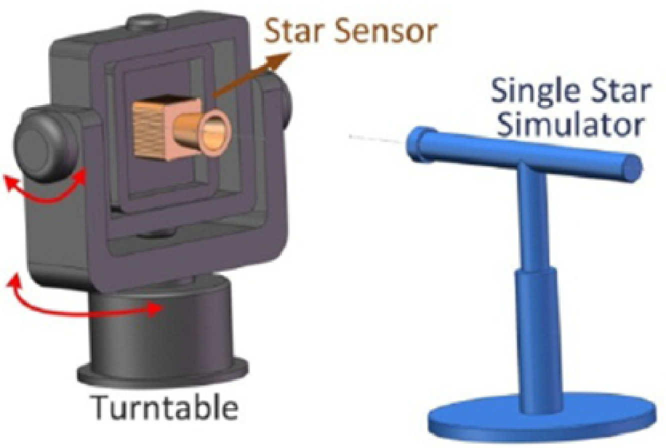


Figure 5.11: Illustration of test setup in a high accuracy star simulator. It consists of the single star simulator collimating telescope and a 2-axis turntable to mount the star sensor and rotate in small rotations to calibrate the off-axis characteristics of the star sensor.

5.8.3.1 Method of testing

A very sturdy and rigid setup which simulated the light coming from a star at infinite distance with respect to the camera parameters was available at the facility. It essentially is a collimating telescope with an adjustable light source at it's focus. The output of the collimating telescope was a parallel beam with extremely low divergence angle.

An accurate rotation table was placed in-front of this collimating telescope on which the *StarSense* was mounted. The rotation table had an accuracy of about $0.1''$. This means that the rotation table could have vibrations in it's set angular position with a standard deviation of $0.1''$. This is negligible as compared to the pixel scale of the camera being tested and thus, would not be detected by the camera. This rotation table allowed a movement of the camera in 2 axes: yaw and pitch. Figure 5.11 shows a schematic block diagram of the setup used for this testing.

The camera position on the rotation table was aligned using a theodolite and the camera optical axis made parallel to the collimating telescope optical axis. Various tests were conducted on the camera to thoroughly evaluate the optical characteristics of the camera.

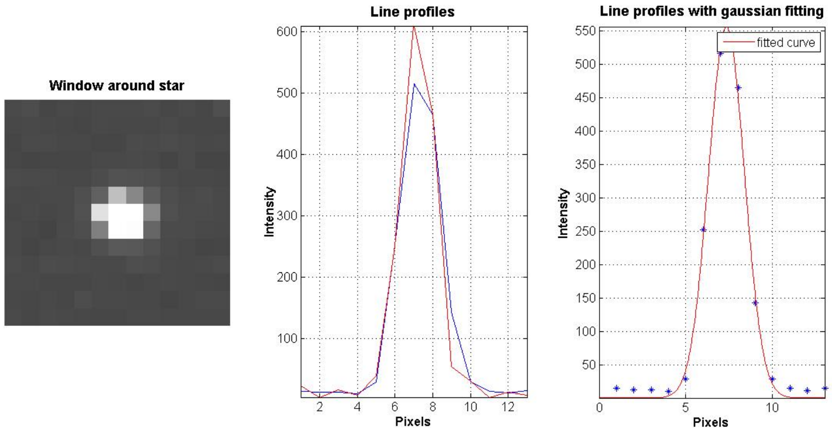


Figure 5.12: In sequence from left to right, 1. Sample point spread function of *StarSense* lens for light from collimating telescope as source, 2. Line profile of the PSF, 3. Gaussian curve fitting the values to estimate the FWHM

The output of the collimating telescope was a plane wavefront simulating the light from a star at infinite distance. The *StarSense* camera is adjusted to image this light on the focal plane as a single point. This point would thus represent the simulated star. The image thus acquired would have a small point spread function due to the diffraction limitation and aberrations of the lens. The lens was designed such that the diameter of this point spread function would be approximately equal to 3 pixels on the detector. Figure 5.12 shows a sample image of this PSF. To obtain the camera geometrical parameters, the orientation of the rotation table was changed with regular angular displacements. The position of the PSF in the resulting image on the detector changed according to the angular displacement introduced and the effects introduced by the lens. The angular displacements provided by the rotation table were very well known. Considering a standard lens model, the estimated location of the PSFs were calculated. The offset from these estimated parameters were due to the lens inaccuracies which could be modeled in various ways. Following set of displacements were given to the rotation table and the images recorded:

- A total angle of ±5 degrees, in steps of 1 degree in Yaw direction
- A total angle of ±5 degrees, in steps of 1 degree in Pitch direction
- A meshgrid formed by sequential displacement of the rotation table in

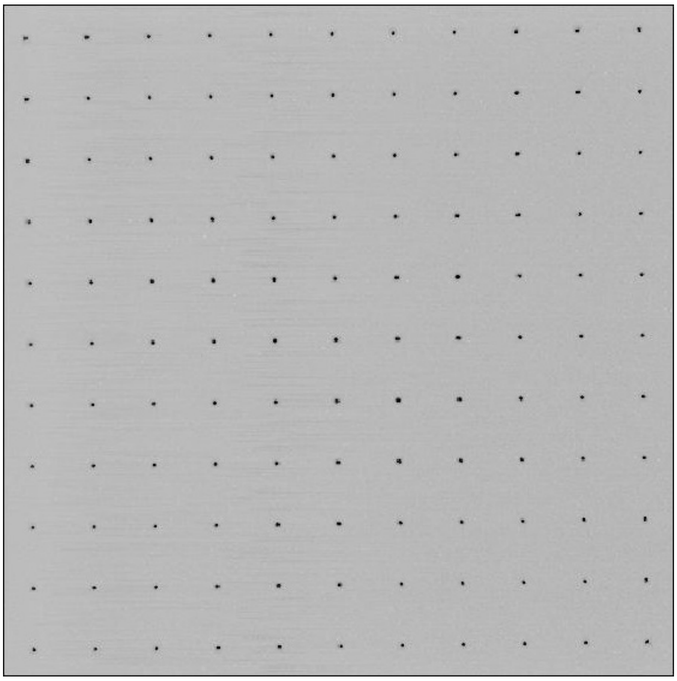


Figure 5.13: Mesh grid of points formed by sequential angular displacements of rotation table in yaw and pitch axes

steps of 1 degree each in Yaw direction followed by 1 degree in Pitch direction. (Refer Fig. 5.13).

5.8.3.2 Results from testing

The above described tests were conducted using accurate instruments and hence were ground reality tests. These were also conducted using the complete imaging system of *StarSense* , thus they evaluated the parameters of the camera considering the consolidated effects of the imaging system viz. geometrical effects, distortions etc. The grid image as shown in Fig. 5.13, is a very useful image obtained as a result from the testing with high accuracy star simulator. This image is generated by combining all the images obtained after each angular rotation of the table. The grid positions essentially give a relation between the angular offsets from optical axis to the linear displacements on the image plane. This can be used to determine various camera parameters like focal length, distortion parameters etc.

In this experiment we have estimated the focal length very accurately using a least squares fitting between the measured angles between the detected cen-

Table 5.9: Intrinsic parameters of camera obtained from testing with high accuracy star simulator

Parameter	Value
Focal length	80.52
Principal point	(544, 626)
Distortion coefficients	-0.08075

troid coordinates and actual values known from the displacement of rotation table. We used the eq. 3.3 to fit the value of focal length to obtain the least rms error between true and estimated angles. At any location in the grid, the focal length can be obtained by using measured angle between the spot at the location and the spots at top, bottom, right and left of the selected spot. This focal length varies over the image plane due to the wide field imaging setup and the real imaging effect caused due to deviation from paraxial imaging. This is modeled as distortion in the camera parameters. We first estimated the mean focal length over the complete field of view using the above mentioned least squares fitting. Further more using the focal length variation obtained over the field we could estimate the distortion coefficients modeling the distortion effect of the lens.

The values obtained after calculation using the data from these tests are tabulated in Table 5.9

5.8.4 Single image of star field to determine camera parameters

Another practical method using the real star images was also used to obtain the camera parameters. In this method we pointed the *StarSense* nominally at zenith by just placing it on the ground and captured multiple images of the star field as the sky passed over it. This experiment was conducted at an observatory where the sky brightness is very low allowing the visibility of many faint stars. The limiting magnitude for *StarSense* camera is $\sim 6^m$. The time, date and location for the observation was noted down carefully and used in a simulation in a software called Stellarium. Stellarium is a planetarium

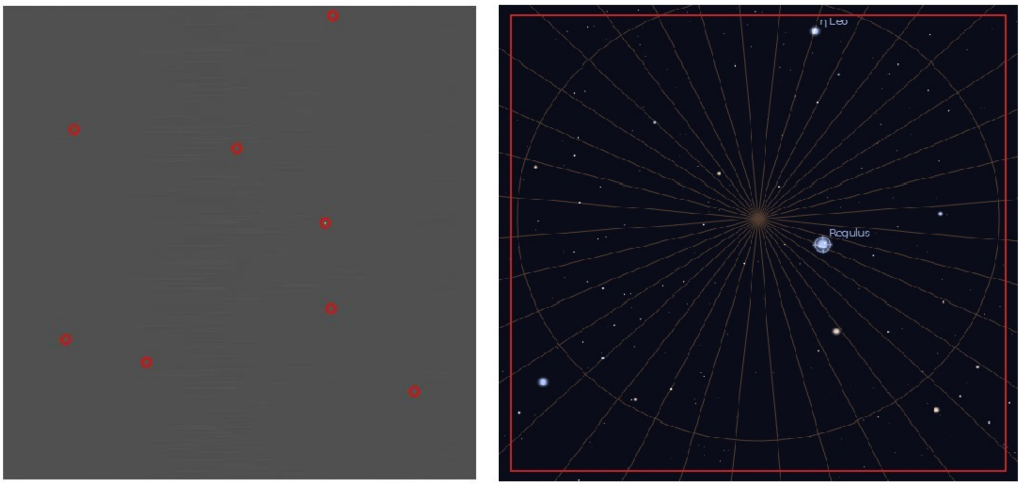


Figure 5.14: *Left* Real image captured with *StarSense* , red marks show the positions of the detected centroids of stars, *Right* Simulated sky image from the software *Stellarium*

software which simulates positions of stars and other objects in the sky as seen from a given geographical location, at a given user input date and time and maps them over on a simulated sky. We simulated the zenith sky image for the observatory location, date and time and matched the stars imaged in the camera with those simulated in Stellarium software. Figure 5.14 shows a sample captured image with detected stars marked, along with the simulated sky image. The right ascension and declination of the stars in ECI coordinate system are noted down. Their unit vectors in ECI coordinate system can be calculated using eq 5.11. Finally the angle between the stars can be obtained using the dot product between the unit vectors from eq 3.4.

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} cos\delta.cos\alpha \\ cos\delta.sin\alpha \\ sin\delta \end{bmatrix} \tag{5.11}$$

where δ denotes declination and α denotes right ascension of the star concerned and (u_x,u_y,u_z) denote the unit vectors of the star in ECI coordinate system.

The unit vectors of the stars imaged in *StarSense* were determined with focal length as a variable parameter. Further the angle between stars were calculated using dot products of these unit vectors. A merit function consisting of the squares of the errors in the angle measurements between simulated angles

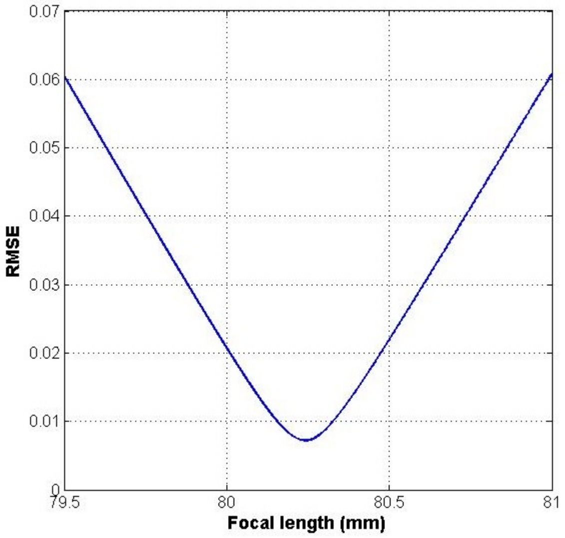


Figure 5.15: RMS angle error between measured angle values and ideal values as a function of focal length of the system

and imaged angles were recorded as a function of focal length. A least squares fitting procedure was used to obtain the focal length of *StarSense* camera. Figure 5.15 shows the plot of the RMS of errors between measured angle values and known angle values for varying values of focal length. The corresponding focal length where this error square value is minimum can be selected as the focal length of the camera. This method of determining the focal length of the camera is especially useful in cases where real-time camera calibration for focal length is required.

5.9 Comparison between various methods of camera calibration

The ZEMAX simulation gives only the focal length estimation and distortion parameters accurately. These are exact measurements of the lens alone and no effect of the structure or assembly is considered in this simulation. It gives an idea as to how many distortion coefficients would be required for correction and where to terminate the calculation in other methods.

The camera calibration toolbox on the other hand gives the values of all the parameters required for calibration. However the method inherently is not very accurate and gives a range of possible values. It is very easy to implement and can be used for a quick measurement of parameters.

The calibration method using the high accuracy star simulator is very accurate in nature and gives all required parameters for camera calibration. But the setup required for the calibration is bulky and costly equipments are required. A special lab is required for conducting the calibration using this method.

The single image calibration method on the other hand can be quickly used for implementation of self calibration in-flight. Focal length being the major factor being affected by vibrations on a rocket launch or temperature variations, this method can be easily used to verify the focal length value and make fine corrections to those value in real-time. Vaz Brendon (2011) gives many other online calibration methods for star trackers and the operational details behind those methods.

5.10 Environmental testing

To qualify the camera for reliable operation in harsh conditions in space and near-space, it is necessary to conduct various environmental stress tests on the camera. Various tests were planned and conducted on the *StarSense* camera to evaluate it's ruggedness and its operation in adverse temperature and pressure conditions. These tests are as discussed below:

5.10.1 Thermal and vacuum testing

One of the primary tests for hardware qualifications is to verify its operation in a low temperature and low pressure environment similar to those experienced in the space flights. In a typical balloon flight, the payload undergoes adverse temperature and pressure conditions. Different material properties of the payload materials can cause failures in the payload operation. In this test, we simulated these adverse conditions experienced by the payload inside a closed chamber. Due to data available from numerous balloon launches earlier, we know ambient temperature profile and pressure profile at higher altitudes. We try to implement an almost similar cycle of temperature and pressure in the chamber and check for any failures in the payload. The following tests were conducted on *StarSense* :

- Thermal test (Cold soak test) of *StarSense*
  - Duration : 4 hours
  - Setup :
    - *StarSense* wrapped in bubble wrap and temperature sensor attached to the body
    - *StarSense* powered from external power supply to be able to measure current at regular intervals of time
  - Temperature and operational current data recorded every 5 minutes
  - It was noted that the current consumption of the *StarSense* reduced marginally during the 4 hour soak test and the sensor remained operational during the complete test. An ambient temperature of about $-20°C$ was reached during the test.
- Thermal test to simulate temperature conditions experienced in balloon flight
  - Duration : 4 hours
  - Setup :
    - Temperature profile experienced in a balloon flight was simulated in a controlled temperature chamber. The profile can shown in Fig. 5.16 was replicated in the thermal chamber.
    - *StarSense* was powered using a battery and kept on during the complete process, logging images to SD card. Time stamps of the images and the images themselves were analyzed after the temperature test.

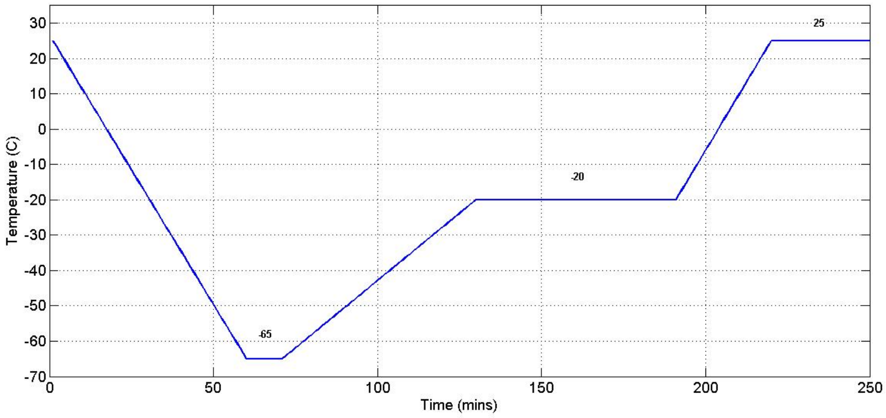


Figure 5.16: Temperature profile simulated in a temperature controlled chamber to simulate the temperature variations experienced in a typical balloon flight. The $-65°C$ is reached at when the balloon crosses the tropopause. Further the temperature increases to about $-20°C$ where the balloon floats for a definite interval of time.

  - It was noted that the *StarSense* continued image acquisition for the whole duration of the test. The timestamps of the images showed regular timings during the whole test procedure. The images captured were analyzed and the dark noise in the image reduced drastically with temperature during the test.

- Vacuum test of *StarSense*
  - Duration: 2.5 hours
  - Temperature: Room temperature
  - Setup:
    - *StarSense* kept in bell jar with controllable pressure
    - *StarSense* kept on with external power supply to measure current variation during de-pressurized condition

The *StarSense* was working even after the low pressure tests. This suggests that the components are sufficiently sealed to avoid damage due to low pressure. The payload drew almost constant current during the complete duration and also the temperature did not shoot up drastically. Hence, it can be satisfactorily assumed that the dissipation surface area of the payload is sufficient to radiatively dissipate the amount of heat generated by the electronics.

5.10.2 Vibration testing

In case of a balloon launch, the payload does not experience too much of accelerations. This is primarily because, the balloon rises with almost a constant speed. The only vibrations it experiences is during the release and the landing. But these values are too low to cause any significant damage to the payload. The flight of the balloon is quite smooth with slow and limited azimuth and elevations variations due to interaction with wind.

On the other hand, a rocket launch is very turbulent. The solid / liquid fuel engines which propel the rocket into space generate a constant sinusoidal vibration in the payload during the launch phase before being deployed into the orbit. These vibrations are significant enough to be absorbed by the payload if the natural frequency of the payload lies anywhere near the sinusoidal vibration frequency. In addition to this, the separation of stages of the launch vehicle generates a jerk / shock in the structure which can propagate to the payload and destroy it completely. Hence, systematic vibration testing of payloads and sensors is mandatory for being launched into space using rockets.

For being qualified to be flown on a satellite as a space payload / sensor, we planned a vibration test to be conducted on *StarSense* . *StarSense* is almost the size and form factor of a cubesat and suitable for being used on cubesats. Hence, the levels to which it should be tested were determined from a vibration test manual for university cubesats. A project QB50 was conducted to study the lower atmosphere of earth using 50 nanosatellites. The systems requirements manual presented for this multi-institution project was a very useful resource to determine various vibration qualification requirements for the *StarSense* (QB50 systems requirement manual , 2014). For vibration testing, the sensor needs to be affixed to a vibration table using required fixtures. These fixtures should be as rigid as possible so as to transfer entire vibrational energy generated by the table to the device under test. The vibration table has the capability to vibrate in one axis, where as the sensor needs to be tested in all the 3 orthogonal axes for vibrations. This requires the fixture to be capable of holding the sensor in all 3 axes. Using such a fixture to mount the sensor in the corresponding axes, following vibration tests are to be performed on the DUT.

- Quasi static test:

  For a quasi-static test of accelerations on the DUT following requirements are to be considered:

  - Direction – XYZ
  - Amplitude – 10.8g

- Resonance survey:

  It is required to run a resonance survey before and after running a test at full level. By comparing the results of resonance survey tests, a change in the DUT integrity due to settling of possible damage can be found. The characteristics required for the resonance survey are as below:

  - Direction – XYZ
  - Type – Harmonic
  - Sweep rate – 2oct/min
  - Start freq – 5 Hz
  - End freq – 500 Hz, extend the frequency to permissible frequency to properly identify the first natural frequency of the DUT and compare with FEA results. (First modal frequency in FEA is 468Hz)
  - Amplitdue – 0.15g

- Sinusoid vibration:

  Following are the characteristics of the sinusoid vibrations tests:

  - Direction – XYZ
  - Sweep rate – 4 oct/min
  - Amplitudes at different frequency ranges :

- 5 - 100 Hz : 2.5g
- 100 – 125 Hz : 1.25g

- Random vibration:

  Following are the random vibration test characteristics:

  - Direction – XYZ
  - RMS acceleration – 8.03g
  - Duration – 120s
  - Power spectral density :
    - 20 Hz : 0.01125 g^2/Hz
    - 130 Hz : 0.05625 g^2/Hz
    - 800 Hz : 0.05625 g^2/Hz
    - 2000 Hz : 0.015 g^2/Hz

The mechanical vibration tests passing criteria was decided as the following:

- The lowest natural frequency of the DUT shall be $>$ 100 Hz.
- Shift in the natural frequency before and after the survey should not be large.
- Integrity of the sensor under all the above mentioned test vibrations

5.11 Conclusion

StarSense being a wide field of view imager, calibration of the camera is very crucial. The importance of the camera model parameters and its knowledge was outlined in this chapter. We discussed and modelled as well as measured various components of noises generated at the detector level. Various tests required to characterize detector noise with respect to varying exposure settings were implemented. The error introduced with varying signal to noise ratio of the imaged source, during centroiding process were also obtained. Also mathematical expressions were obtained to relate the signal measured in the detector with the real star magnitude. Thus, a basic photometric calibration was obtained for the camera. This photometric calibration was done both in laboratory as well as real sky conditions.

In addition to all the detector calibration and sensitivity tests, various camera model parameter calibration tests were undertaken. The importance of these parameters in the star sensor algorithms and the process of data reduction from images to quaternions was shown. Various methods to obtain these parameters and validate them were highlighted and a comparison was shown between the measured values using all of these methods. This comparison helped in selecting the most suitable and accurate method for the calibration of *StarSense* . While designing various other star sensors, similar methods can be used to obtain their camera model parameters and validate their design.

To ensure that the camera assembly would work in adverse environmental conditions in space environments, various environmental tests for temperature and pressure were undertaken. In addition to ensure the integrity of the sensor during the launch of the spacecraft, a vibration test plan was proposed for testing of the sensor. This would also validate the mechanical design of the housing and its correctness.

After obtaining the camera calibration parameters and ensuring its effective operation in adverse environment conditions, it was ready for functional validation using star patterns in a lab as well as in real sky conditions. These functional validation methods and results are discussed in the following chapter.

Chapter 6

Functional validation of *StarSense*

6.1 Introduction

A star sensor is crucial sensor in the attitude control system of a high altitude balloon based observation platform or a space telescope. It plays an important role to determine the current pointing direction of the telescope with respect to reference coordinate system. It plays a critical role in the desired operation of the payload and to help steer it in desired direction. The output of the star sensor is given to the pointing control system which calculates the error between the current pointing direction measured by the star sensor and the set value of the pointing direction. A proportional, integral and derivative (PID) control action based on the magnitude of error is then passed on to the actuators of the high altitude balloon platform or the satellite to correct the error. In case the attitude information given by the *StarSense* is erroneous, the outputs generated by the control system would be erroneous resulting in random unpredictable behavior of the control system. This would result into failure of satellite/ platfor operations and failure to capture crisp and clear images of the target source by the telescope or worser still complete loss of connectivity due to communication channel failure. Thus, a star sensor failure results in the complete loss of functionality of the spacecraft.

Considering the importance of the desired operation of the star sensor, it is imperative to carry out various functional tests on the star sensor to verify and validate its functionality and identify and remove the glitches it might face. The glitches experienced in the testing phase are very useful to design a robust flight software which will rugged enough to avoid all these glitches. Also a characterization of the functionality of the sensor is thus performed. There are two types of errors when considering the accuracy of the star sensor: the steady state error and jitter. Both of these need to be measured to characterize the star sensor completely.

In this chapter we describe various functional validation tests carried out on *StarSense* , their results and the conclusions we derived from this information.

6.2 Conversion of quaternion to reference coordinates

For functional validation of the sensor, it is necessary to orient the sensor in a reference direction and identify the direction by the measurements from the star sensor. The performance of the sensor can be evaluated by comparing the calculated quaternion with the quaternion corresponding to the reference direction. We carried out this performace evaluation using a test setup in a lab and a test setup with a telescope. In both cases, the reference quaternion for the star sensor was provided in the form of inertial coordinates defined in the software or as pointing direction of the telescope. Hence, to validate the sensor output, it is necessary to either convert the output quaternion of the sensor into inertial coordinates to compare with the input set values or vice-versa. To evaluate the correctness of quaternions, a meaningful visual representation of the quaternion is required. The process of this conversion is described further.

There are 2 steps to convert the quaternion into meaningful inertial coordinates. The quaternion is essentially a representation of the rotation between two coordinate systems where as right ascension and declination is a representation of the direction of a given unit vector in the inertial coordinate system. To obtain the right ascension and declination corresponding to a certain quaternion essentially requires calculating the unit vector in inertial coordinate system given the unit vector in sensor body coordinate system and the rotation matrix of the inertial coordinate system with respect to body coordinate system as described in eq. 6.1. In terms of quaternion representation this rotation process can be represented as shown in eq. 6.2 where $\mathbf{q^*_{ib}}$ is the conjugate quaternion. The quaternion value is obtained from the star sensor after realtime processing of sky images. In this case we consider the unit vector direction of the bore sight of *StarSense* . From the definition of the sensor body coordinate system, the bore sight of *StarSense* is defined as the z-axis ($\mathbf{v_b} = (0, 0, 1)$) of the sensor body coordinate system. The next step is to calculate the right ascension and declination of the unit vector direction from it's definition. We know from eq 6.3 that the unit vector in inertial coordinate system is defined as a function of the right ascension and declination of the pointing direction of the unit vector and can be obtained as shown in eq 6.4 and eq 6.5. This conversion was used to compare the estimated quaternion with the reference quaternion used in the test setup.

$$\mathbf{v_i} = R_{ib} \times \mathbf{v_b} \tag{6.1}$$

$$\mathbf{v_i} = \mathbf{q^*_{ib}}.\mathbf{v_b}.\mathbf{q_{ib}} \tag{6.2}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} cos\delta.cos\alpha \\ cos\delta.sin\alpha \\ sin\delta \end{bmatrix} \tag{6.3}$$

$$\alpha = tan^{-1}\left(\frac{v_y}{v_x}\right) \tag{6.4}$$

$$\delta = sin^{-1}\left(v_z\right) \tag{6.5}$$

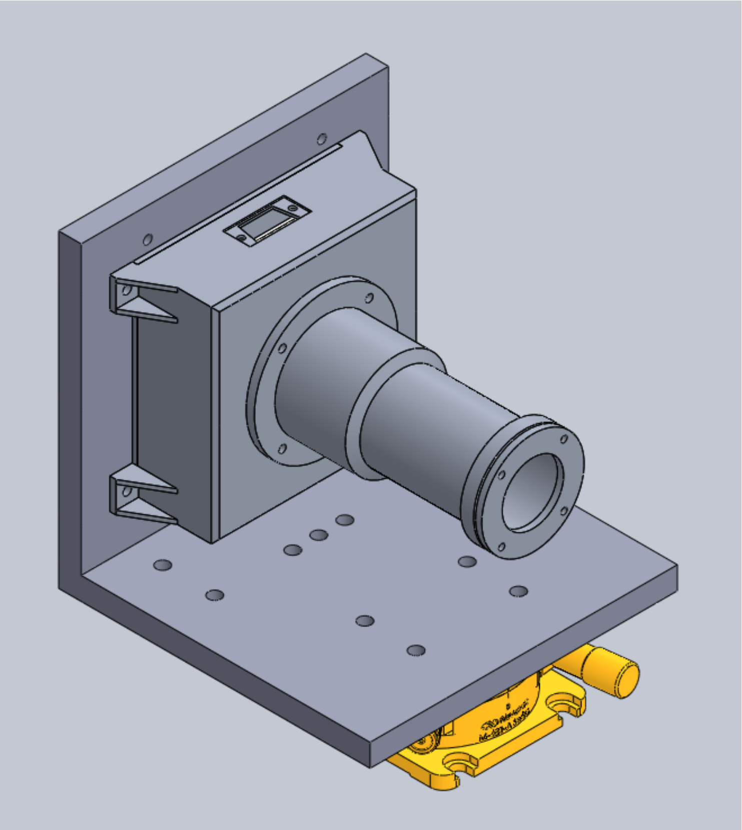


Figure 6.1: Integrated CAD model of mounting platform and the rotation stage used to mount the *StarSense* on an optical table

6.3 Test setup hardware

For the testing of the star sensor, it is necessary to mount it rigidly on an optical table along with the other apparatus used for testing. To enable mounting it on the optical table, we fabricated an 'L' section. Arrangement was made to mount the complete assembly on a rotary stage also to be able to adjust the orientation of the sensor with respect to the collimator source. Figure 6.1 shows an illustration of the 'L' section mount platform.

6.4 Dynamic multi star simulator

6.4.1 Test setup and methodology of testing

For the first level of testing, an elaborate test setup in the lab was used to simulate star field patterns. Fig. 6.2 shows a layout of this test setup. For the functional validation of the star sensor in a lab, a setup was required to simulate star field and patterns of stars similar to those visible in the sky. To achieve this, a setup as shown in figure 6.2 was used. The LCD screen controlled from a computer was mounted rigidly on the optical table. An adjustable neutral density filter was mounted in front of the LCD screen. A collimator lens was mounted in front of the filter in such a way that the LCD screen lies at the focal plane of the collimator lens. The LCD displaying star patterns being at the focal plane of the collimator lens, generated a plane wavefront at the output of the collimator and simulated stars at seemingly infinite distance from the *StarSense* . The collimator lens used was a custom designed lens which corrected especially for the off-axis aberrations and field curvature considering a large planar focal plane for the collimating lens. The size of the LCD screen and focal length of the lens was such that it simulated a total field of 25 deg of sky for the star sensor. The field of view of *StarSense* being only 12 deg, it was easily accommodated in the simulated star field.

A software on the computer connected to the LCD screen allowed the user to select a particular portion of the sky by entering the coordinates of that portion through a user interface. Various dynamic conditions simulating the slewing of the satellite with different angular speeds could also be simulated in the software. Hence, the test setup was called Dynamic Multi Star Simulator (DMSS).

To mount *StarSense* in-front of the optical layout, a rigid mount was required. The 'L' section mount was placed on top of the optical table with coarse height adjustments and coarse alignment of the optical axis of the layout and *StarSense* . A fine adjustment of the alignment and height was done in software during image processing which is as explained further.

6.4.2 Diamond pattern fine alignment

To determine fine alignment of the star sensor with the optical axis of the validation layout, an image with a diamond pattern was displayed on the screen (Fig. 6.3). The center star in the pattern indicated the optical axis of

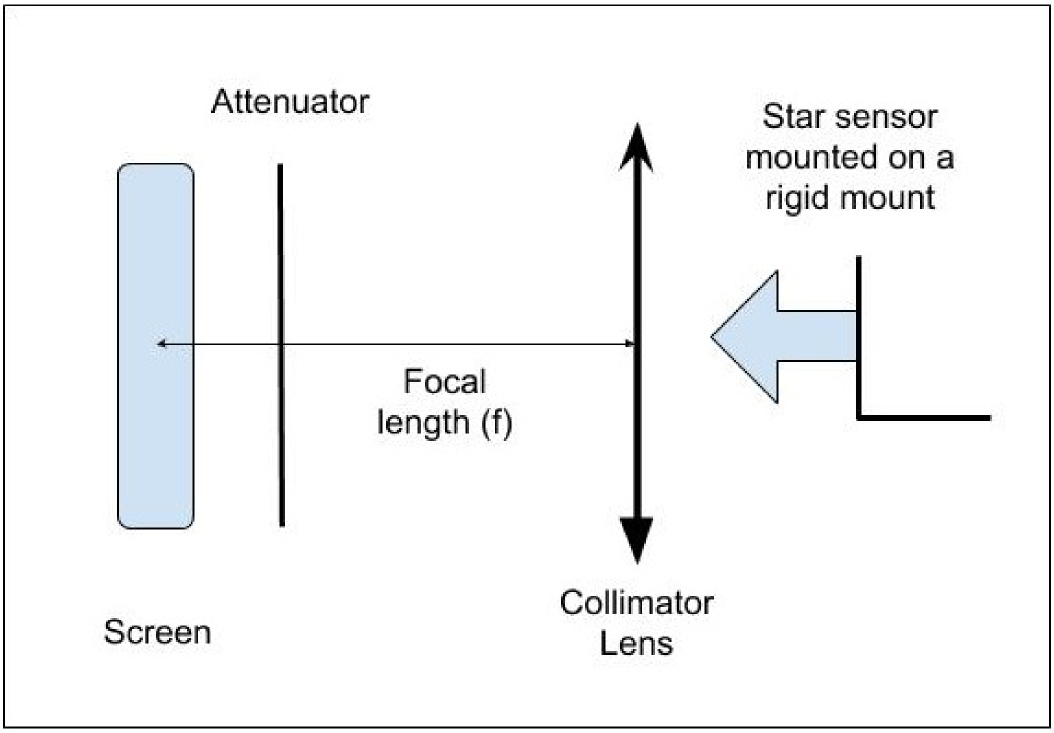


Figure 6.2: Layout of components used for functional validation of *StarSense* in a laboratory.

the test setup layout. 3 ways of correcting for the misalignment were identified. 1. Fine adjustments using micrometer screws and rotation tables, 2. Correct for offset between center star centroid coordinates in the captured image with respect to the principal point of the star sensor. 3. Correct for the offset in terms of rotation matrix/ quaternion after calculating the offset quaternion from the calculated offsets in the right ascension and declination directions respectively.

Considering the unavailability of hardware in the restricted laboratory, we assorted to software corrections for misalignment errors. The centroid of the center star of the diamond pattern was determined from the initial images. This location was used to determine the optical axis of the collimator assembly. During processing of images, each centroid coordinate was shifted an amount equal to the offset on both the axes of the image sensor. This process was used to manually correct for the misalignment in the two optical axes.

Another similar method of achieving this was identifying the quaternion distance corresponding to the offset shifts on the two image sensor axes. This

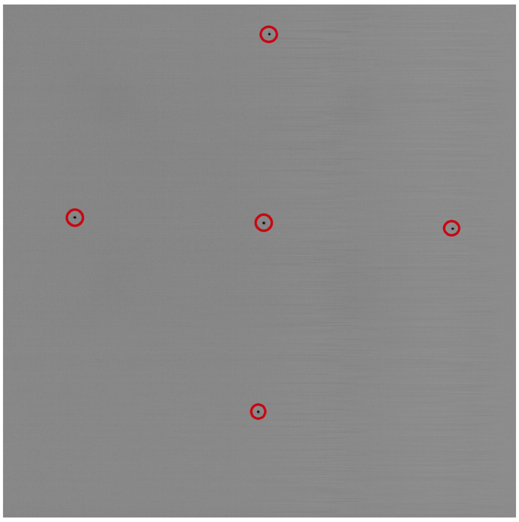


Figure 6.3: Diamond pattern displayed on screen used to align *StarSense* with the optical axis of the test setup. The middle point in the diamond pattern defines the location of the optical axis of the test setup.

quaternion was eventually multiplied with the obtained final quaternion from QUEST algorithm from the *StarSense* .

6.4.3 Static pattern test

After having co-aligned the optical axes of the collimator assembly and *StarSense* , a static pattern test was conducted. In this test, a static star pattern corresponding to a user configurable sky field was projected by the LCD screen. The right ascension, declination and roll angle of the camera with respect to the ECI coordinate system entered by the user selected the star pattern. These angles served as Euler angles for the rotation of the camera. The LCD screen was controlled from a computer outside of the dark room where we could enter these user parameters. The brightness of the simulated stars were also adjustable using an attenuation scaling parameter. The static pattern test was conducted at random directions in the sky and a time series of quaternions were recorded over certain time intervals for each random direction. The measured orientation from the *StarSense* was converted into inertial coordinates, compared with the reference values entered by the user and the error values were recorded.

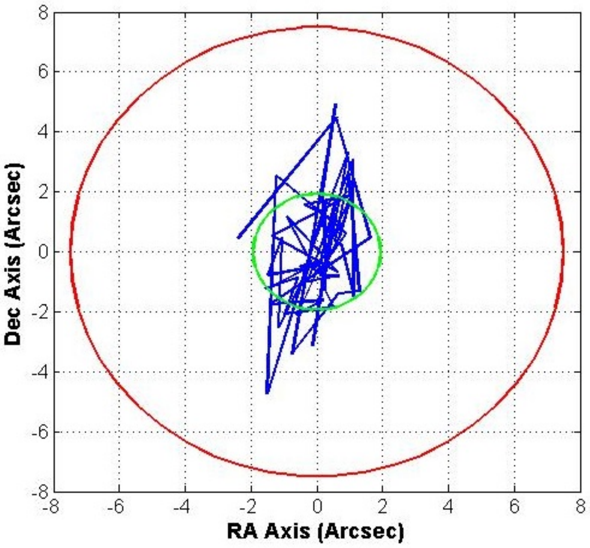


Figure 6.4: Right ascension & declination error diagram to show the spread of the measured value with respect to the reference value. The target spread is marked by the red circle whereas the achieved spread is marked by the green circle.

6.4.4 Results and discussion

The quaternions obtained from the star sensor were recorded in the form of a time series. Eq. 6.2, 6.3, 6.4 & 6.5 were used to obtain right ascension and declination from the quaternion of the star sensor. The unit vector used for conversion was the z axis of the sensor coordinate system which corresponds to the boresight of the sensor. These right ascension and declination were compared with the reference right ascension and declination which was obtained from the input value given to the DMSS control computer to be displayed on the LCD screen. The error was plotted as a circle diagram as shown in Fig. 6.4. The red circle in the figure shows the required accuracy of the star sensor which was its designed value. The green circle shows the measured accuracy of the star sensor which is the standard deviation of the obtained error values. This shows that the star sensor performs better than the expected design accuracy value. After having obtained good results in lab conditions, the star sensor was also tested in real sky conditions using a telescope co-alignment method.

Table 6.1: Test setup of the star sensor coaligned with the telescope and the functional validation being carried out at Kavalur Observatory.

6.5 Telescope co-alignment test

6.5.1 Test setup

The *StarSense* was co-aligned with a tracking telescope in an astronomical observatory, which allowed to determine the sensor response in real sky conditions. A mounting plate was designed and fabricated to mount the *StarSense* coaligned with the telescope. This plate was assembled in place of the view finder of the telescope. The field of view of the *StarSense* being very wide as compared to the telescope (0.1°), coarse alignment of the *StarSense* with telescope was acceptable. The error in the measured quaternion with respect to the reference quaternion would remain constant during the coarse of the experiment and could be isolated at the beginning of the experiment. The setup was arranged as shown in fig. 6.1.

6.5.2 Methodology of testing

Mounting of the star sensor on the telescope on a rigid plate gives an ability to co-align the telescope with the star sensor. Further, the telescope was pointed at different directions using the control center of the telescope by entering the right ascension and declination values for the pointing direction. The telescope was pointed at multiple directions selected randomly in the sky visible on that day. The visibility of different brightness stars in the sky was dependent on the exposure settings (integration time and gain value) of the *StarSense* camera. Images were captured at varying exposure setting to finally obtain an optimum exposure setting for visibility of maximum number of stars with an integration time of 0.2 seconds and 4x gain.

6.5.3 Results and discussion

At each pointed direction of the telescope, several images of the sky were captured and quaternions obtained from these images. These quaternions were then put together as a time series. The telescope pointing direction being very stable of the order of $0.1''$, would not be noticed in the quaternion calculation error as seen from the image. The quaternions obtained were then converted in the right ascension and declination values to compare with the input right ascension and declination values which were fed into the telescope control software to point the telescope. This was done using Eq. 6.2, 6.3, 6.4 & 6.5. The error values between measured and actual quaternion were plotted on a circle diagram as shown in Fig. 6.5 Thus, this test essentially validated the functionality of *StarSense* in real sky conditions.

6.6 Quaternion accuracy degradation with sensor movement

In presence of the disturbances experienced during a high altitude balloon flight, the images captured by the star sensor in a long exposure of 100-200ms get blurred due to angular motion of the camera. This has mainly 2 effects: the SNR of stars reduces, resulting in fainter stars not being detected and generation of trails in the star images. Considering the star sensor to be a

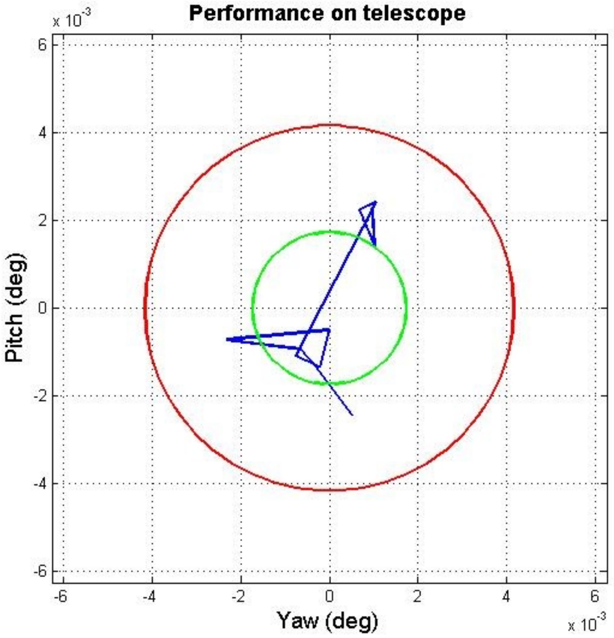


Figure 6.5: Right ascension & declination error diagram to show the spread of the measured value with respect to the reference value. The target spread is marked by the red circle whereas the achieved spread is marked by the green circle.

very accurate orientation measurement device, the second effect would cause a major problem in it's operation as it would lead to degradation in quaternion accuracy. A separate analysis was done to determine the suitability of the developed sensor for balloon platforms. As observed from the balloon experiment, the balloon platform experienced angular rotations of about 0.5°/ sec at floating altitudes whereas the disturbances were much higher during the ascent stage. Thus, it is not wise to use the star sensor as an attitude sensing device during the ascent stage, but after reaching floating altitude, it can be successfully used in the pointing control system. An image simulation was done to estimate the degradation of the quaternion accuracy at various angular speeds.

In this simulation, a star field was randomly selected from the Hipparcos star catalog to simulate the field of view as seen by the star sensor. A single image of the star field was generated by integrating several smaller exposures with exposure times of about a millisecond. The motion of stars during this millisecond sub exposure was assumed to be negligible, where as only the motion between the millisecond sub exposures was considered. The stars in the selected star field were projected on to the image sensor plane first and their

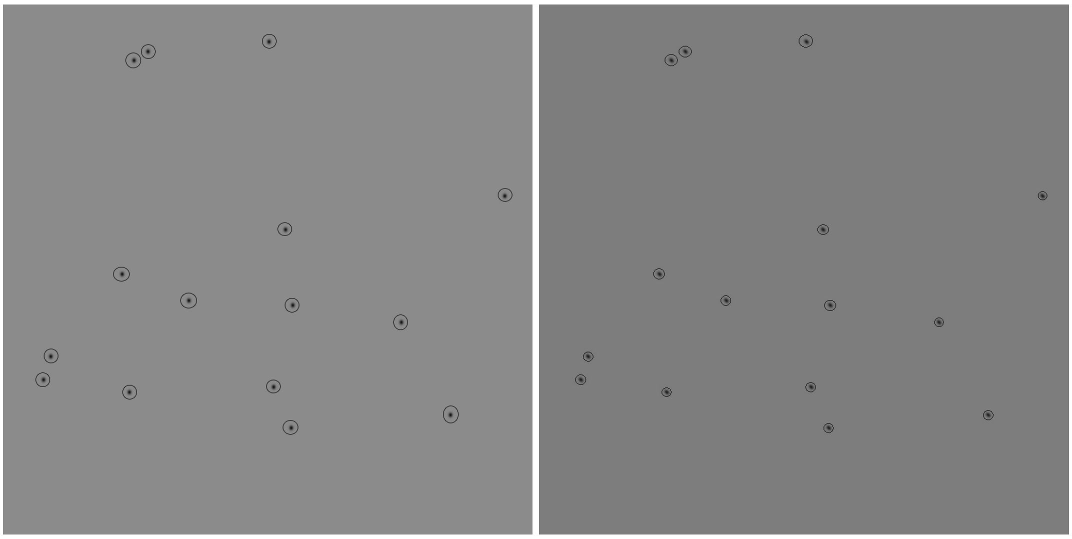


Figure 6.6: Simulated images under 0.5°/ sec slew rate of camera in pitch and roll directions respectively. These simulations were carried out considering integration of an image using smaller sub exposures of 1 millisecond each assuming a stationary centroid coordinate of stars during the sub exposure.

initial centroid pixel coordinates were determined from the camera geometry. A 2 dimensional Gaussian function was used to describe the intensity distribution of the stars on the image in each sub exposure. The FWHM of the Gaussian function was assumed to be equal to the FWHM of the PSF of the *StarSense* optics. Every consecutive sub-exposure was generated by shifting the centroids of the stars in the image sensor by an amount corresponding to the angular shift during the sub-exposure. In this case the incremental change was obtained considering the angular speed of 0.5°/ sec and a pixel scale of 10°/1024. The generated images displayed streaks of light due to movement (as shown in Fig. 6.6). This reconstructed image simulated the images captured by the star sensor when it was undergoing angular motion. The star sensor algorithms were applied on these images and the orientation direction determined. The error with respect to the initial selected direction was noted.

These images when generated at a range of angular speeds allowed us to determine the degradation in the attitude sensor output in conditions where the *StarSense* was rotated at different angular speeds. This was done by generating 100 images at each angular speed. The standard deviation of the quaternion output was plotted as shown in Fig. 6.7.

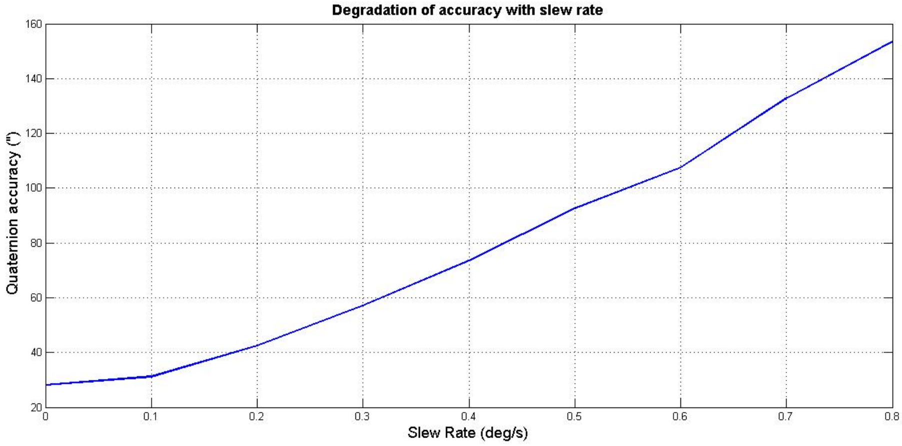


Figure 6.7: Star sensor accuracy degradation with slew rate

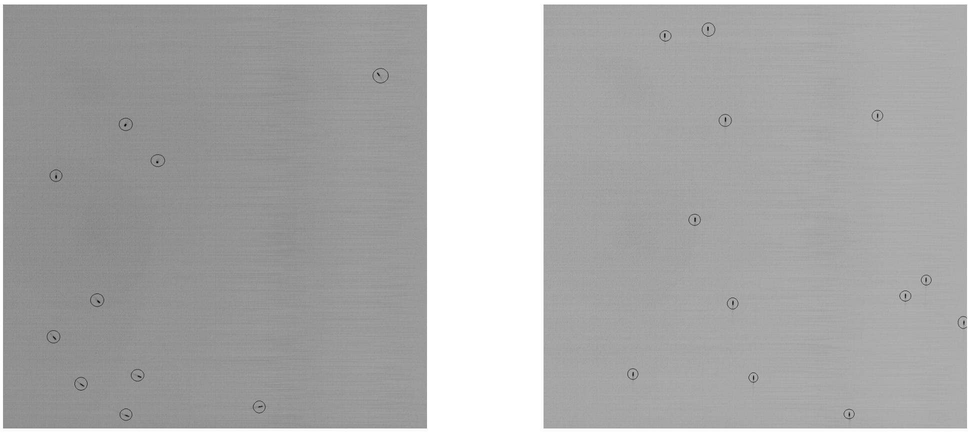


Figure 6.8: Images of the simulated dynamic sky captured at DMSS, LEOS. *Left* Image captured at an angular speed of 0.5°/ sec in roll axis, *Right* Image captured at an angular speed of 0.5°/ sec in pitch axis

In addition to the simulation, an image test was done at LEOS DMSS to determine the images obtained under rotating conditions at different angular speeds. The captured images are as shown in Fig. 6.8. The streaks as seen in the image are caused due to significant angular displacement of the stars in the simulated sky during the long exposure which is necessary to get desired SNR of the faint stars.

6.7 Conclusion

After careful calibration of the *StarSense* camera, we also validated the functionality of the image processing algorithms on the images captured by the

StarSense . We used standard test setups for functional validation like the dynamic multi star simulator to simulate real sky conditions in laboratory. It allowed synthetically pointing the *StarSense* to different right ascension and declination with different roll angles. It also facilitated slewing of the camera in any direction with adjustable speed. Further more we tested the *StarSense* by coaligning it with a telescope and pointing the telescope very accurately towards a certain direction. This experiment allowed simulating real sky conditions during balloon flights much more closely. The accuracy of the star sensor obtained under static conditions was measured to be much better than the target value and thus our original requirement of $30''$ were met.

In addition to static functional validation, the slew capability in the dynamic multi star simulator allowed validation of the accuracy degradation estimate which we had predicted. With a slewing camera, the attitude accuracy would reduce based on the slew rate of the camera. After all of these validations, the *StarSense* was deemed suitable for use on balloon observatory. We considered the typical balloon slew rates of the order of 0.5° where the accuracy of the sensor in worst conditions would degrade to about $80''$ which is finer than the $\sim 0.25^\circ$ accuracy offered by attitude sensor.

Chapter 7

Conclusion and future work

7.1 Balloon platform and typical disturbances experienced

High altitude balloon platforms are a lucrative, low cost and short time-line way of generating useful scientific data related to ultraviolet astronomical sources. They are specifically useful for observing sources which come at a very short notice and require a rapid response such as comets, transient events etc. Telescopes mounted on these high altitude balloon platforms provide significant advantage over operations from ground, enabling observations at forbidden wavelengths. A UV telescope (200-400 nm) in stratosphere, with an aperture of 6 inch in diameter and sufficient pointing stability/ accuracy and a $1K \times 1K$ CCD array can provide wide field images with FWHM better than 100 approaching the diffraction limit (Fesen et al., 2015), similar to that of space observatories but at much lower cost. As mentioned previously in chapter 1, in the high altitude balloon program initiated at Indian Institute of Astrophysics, we carry out similar experiments which involve sending a telescope along with a spectrograph payload or an imaging payload to stratospheres. As described previously, the pointing system being developed to allow stable/ accurate pointing of the telescope to desired astronomical sources requires sensors to measure the attitude of the platform in real time. It is necessary to understand the disturbances experienced by the telescope platform at high altitude due to various factors like wind, thermal and pressure differences etc and preliminary experiments were conducted in this direction to establish the same.

Stratospheric balloon platforms essentially consist of a balloon train as described in chapter 1. Besides transmitting the balloon's buoyant force, the flight train is the source of disturbances that the pointing control system must reject. A typical stratospheric balloon has a train of several meters in length comprising of a recovery parachute which ensures safe landing of the payload platform. This is connected to a gondola containing scientific equipment with communication and associated electronics. Similar flights are conducted at Tata Institute of Fundamental Research at the National Balloon Facility at Hyderabad. On one of their launches conducted on 8th May 2016, we got an opportunity to mount an inertial measurement unit sensor, to evaluate the natural motion of the gondola during the balloon flight. This allowed to characterize these natural disturbances experienced by the gondola and in-turn the payload residing inside the gondola.

The data collected in the stratospheric balloon flight by our inertial measurement unit sensor and the support data obtained from TIFR combined together provided considerable insight about the wind disturbances at different altitudes during the flight. The altitude variation of the balloon against time during the flight is displayed in Fig. 7.1. We can see clearly that the balloon reached a floating altitude of $\sim$ 30 km at around 2 Hrs from the launch.

Fig. 7.2 also shows the angular position of the IMU in azimuth direction and also wind speed experienced by the balloon. The cylindrical graphs depict the azimuthal direction of the payload with respect to ground North as marked. The time axis is shown in the 3rd dimension. The variation of this azimuth after 2 Hrs from time of launch, which was the time when the payload reached floating altitude, is reduced considerably. Thus, it can be seen that at floating altitudes, the payload platform becomes stable and experiences lesser

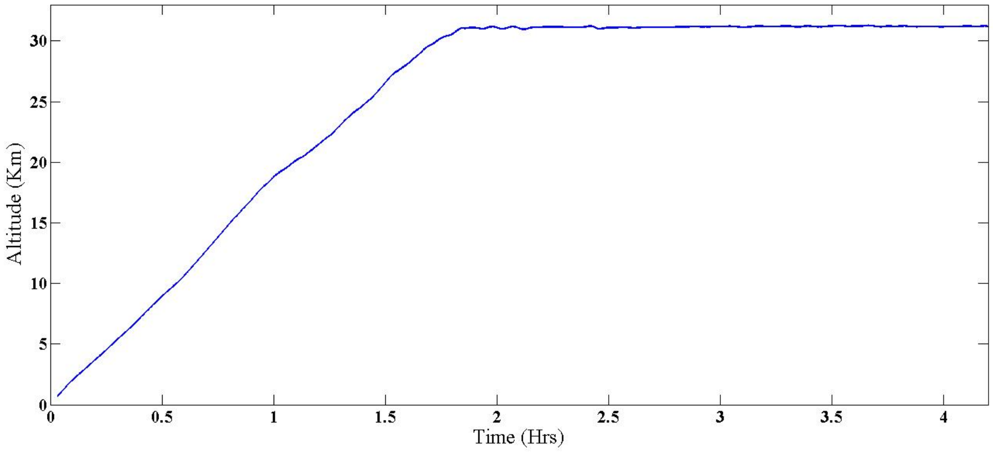


Figure 7.1: Datalog of altitude of the balloon with time starting from launch. The balloon floats at an altitude of about 30km which is evident from the graph.

disturbances.

The inertial measurement unit used in this balloon launch was a very useful sensor. It contained a digital motion processor which allowed computation of Euler angles of the balloon payload by using the accelerometer, gyroscope and magnetometer data. Out of this, essentially azimuth and elevation data was important. As mentioned earlier, the reduced disturbances at floating altitude were noted from the azimuth, but in addition to that it was noted that the angular speeds experienced by the payload ranged to an order of about $0.5^\circ/\,\text{sec}$.

7.2 Conclusion

Considering all the above mentioned motions of payload on a balloon platform, a pointing system is rendedered imperative for astronomical observations. Also various stability requirements of the pointing system can be identified from the analysis done above. A simple pointing system which can achieve an accuracy of $\sim 0.25^\circ$ is discussed in Nirmal et al. (2016). To design a balloon observatory platform for near UV observations as discussed in chapter 1, more sophisticated sensors and actuators were required. In this thesis we have discussed the design, development, calibration and validation of one of the sensor necessary

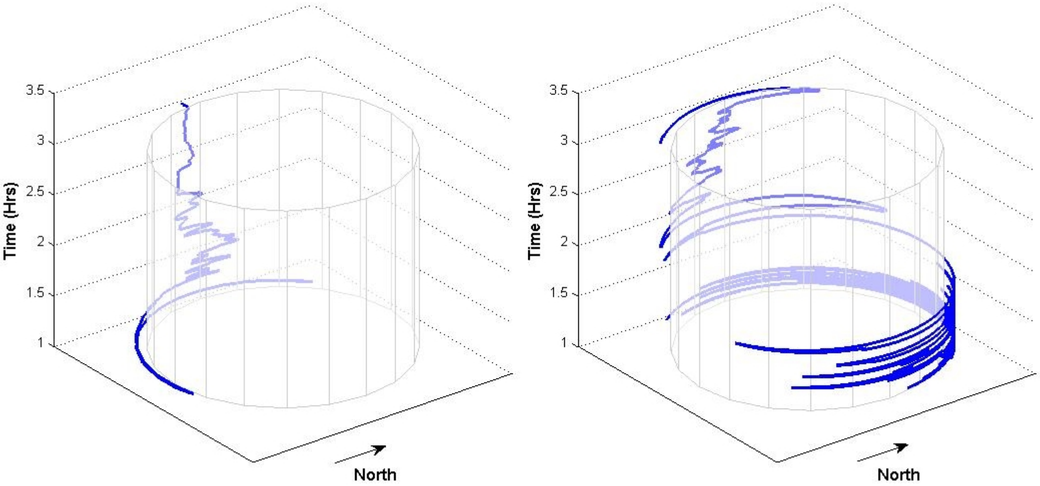


Figure 7.2: Wind direction (*Left*) and movement of the payload in azimuth (*Right*) after reaching the stratosphere. Time axis shows time in hours elapsed since launch. The float altitude was reached at about 2 hrs after launch. The plots clearly show the difference between the rapid variation of payload azimuth before reaching the final float altitude and less variation of payload azimuth after reaching the float altitude.

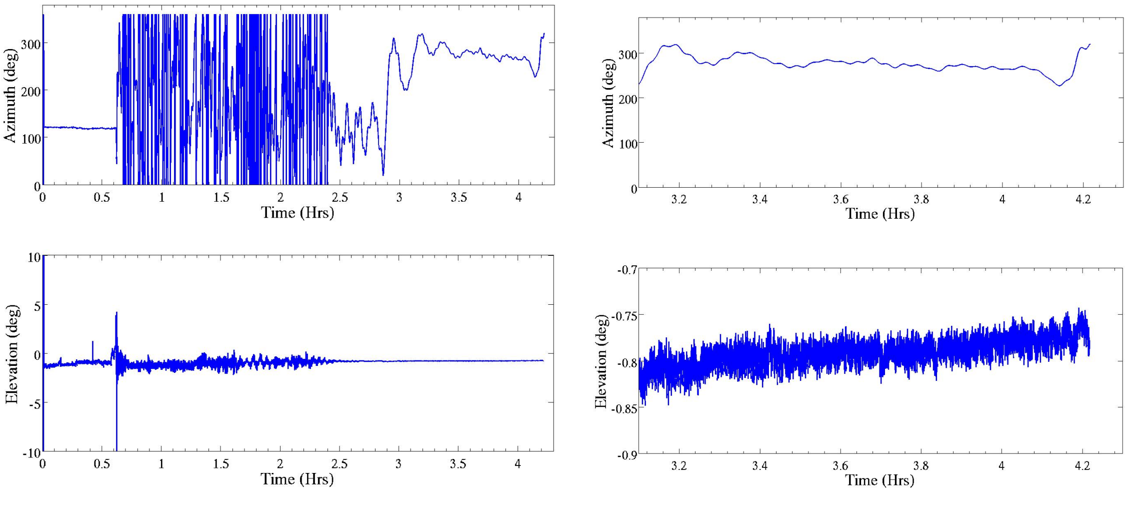


Figure 7.3: Movement of payload in azimuth (yaw) (*Top*), and elevation (pitch) (*Bottom*). X-axis is time in hours from the switch on at 5:35 am IST. On the left is the total retrieved data, and on the right is the data at the float altitude.

towards building this balloon platform.

We begin with the description of the requirements for this sensor as an optical device. From the understanding of the bright star distribution in the sky we have identified the field of view and limiting magnitude requirements for a star camera. These values served as target values for first order design of the optical system of the sensor. Further we analyzed various algorithms used on the sensor to compute the required output quaternion. Various algorithms like centroiding, geometric voting and QUEST algorithms were prototyped, fine tuned and implemented first in the form of a MATLAB package and later as a firmware on an embedded system. The performance of these algorithms when used over different directions in the sky were also considered and evaluated. Various parameters like memory requirements, timing summary for algorithm implementation, sky coverage and accuracy were estimated for these algorithms. This enabled a virtual evaluation of the star sensor implementing these algorithms even before it was even built.

A complete block diagram of the sensor was conceptualized to make it a modular system to be integrated later. The lens system used in the sensor was designed for a wide field application with minimized distortion, coma and other off-axis aberrations. The design and tolerance analysis done before fabricating the lens system is discussed in the thesis. The electronic subsystem worked as a core of the functionality of the sensor. The architecture of the electronic subsystem, various design decisions etc are discussed there after. This electronic subsystem is designed in a generic way so as to use it as a backend of other sensors. Another version of this star sensor with a different image sensor was built based on the same backend processing subsystem. It was also used in other payloads such as described in Mathew Joice et al. (2017) and S. Ambily (2018). To house all the components and make the sensor suitable for adverse environments in space, a housing was designed. Rigorous finite element analysis was done on the housing to understand it's suitability for launch on balloons and space. This included various stresses imposed on the housing system to check different natural modes of vibrations excited in the housing as well as failures if any occured. Also for making the sensor suitable for use in space, a baffle was designed which would avoid stray light entering the system from sun and earth albido. In cases of balloon launches this baffle was not necessary as the launch would usually take place at night.

Calibrating the sensor after integrating different subsystems of the sensor is a necessary procedure for proper reliable operations. Different characteristics of the sensor had to be calibrated like the noise characteristics seen in the detector, camera calibration parameters etc. These calibration tests were carried out in multiple ways which included ways to calibrate these parameters online too. The results obtained were validated with design values of the sensor as well as standard calibration procedures. This evaluated the effectiveness of the online procedures we had used to calibrate the sensor. These calibrations procedures and results are also discussed in detail in this thesis.

Functionally validating the sensor is of utmost importance before the sensor can be used in a control system. We tried a new validation procedure by using the sensor in the field, mounted on a telescope and testing it's effectiveness. The results were compared with those from the standard laboratory method of calibration. These validation procedures were also discussed in the thesis.

Considering all the analysis and design of the *StarSense* already done, we could quickly modify it and make it suitable for applications on a cubesat too (Figure 7.4). We could achieve the size reduction in the *StarSense* camera to make a $\mu StarSense$ mainly by reducing the pixel pitch of the image sensor from $15\mu m$ to $4.8\mu m.$ The reduced size of image sensor required a much small focal length lens to cover a 12° field of view thus cutting drastically into the length of the system. This reduced size sensor could now fit in a 0.5U cubesat which was suitable for >3U cubesat applications. For the $\mu StarSense$ we used an off-the-shelf ruggedized lens system from Schneider Optics which was tested earlier in industrial environments with large vibrations. As a replacement of the Star1000 image sensor, we used an industrial grade image sensor with a large window for operating temperature range. The backend electronics used in this sensor was repurposed from the original *StarSense* , thus proving the suitability of the generic FPGA board to be used in multiple applications. The image sensor card being separate from the FPGA card in the electronics subsystem, we could make a new image sensor card very quickly and use it in the *μStarSense*. Various specifications of the *μStarSense* are tabulated in a user guide shown in Appendix. Also the calibration of this *μStarSense* is done in a similar way as explained in chapter 5. The test results of the calibration are also shown in Appendix.

In addition to using it as a star sensor, the *μStarSense* was capable of capturing images from sky. It could also be used as an earth observation camera from the cubesat platform. With the selected optical and electronics components we analysed that it could have a ground sampling distance of about 100m from an orbit at altitude of 600km. This was very much comparable to some of the off-the-shelf cameras available for earth observation from cubesats like the GOMSpacce Nanocam[1]. To consider for the limited capability to transfer images to ground stations with limited data rates and visibility times, it was imperative to implement compression techniques on the camera. We implemented a JPG compression verilog IP to compress the stream of incoming pixel data from the image sensor. The reconfigurable property of FPGA based embedded system allowed addition of such peripherals to the camera. We provided a standard I^2C interface on the camera to make it compatible to the on-board computers used on cubesat platforms. This allowed us to target the utility of the *μStarSense* on a university satellite platform called PISAT from PES Institute of Technology through technology demonstration collaborations.

7.3 Future work

The *StarSense* being an optical camera on satellites with a capability of online image processing, it can be used to implement photometry in real time applications to draw light curves of stars being observed. Being sensitive to bright stars upto 6^m, it can be used for sensing the light curves of the stars while it

[1] https://gomspace.com/shop/payloads/earth-observation.aspx

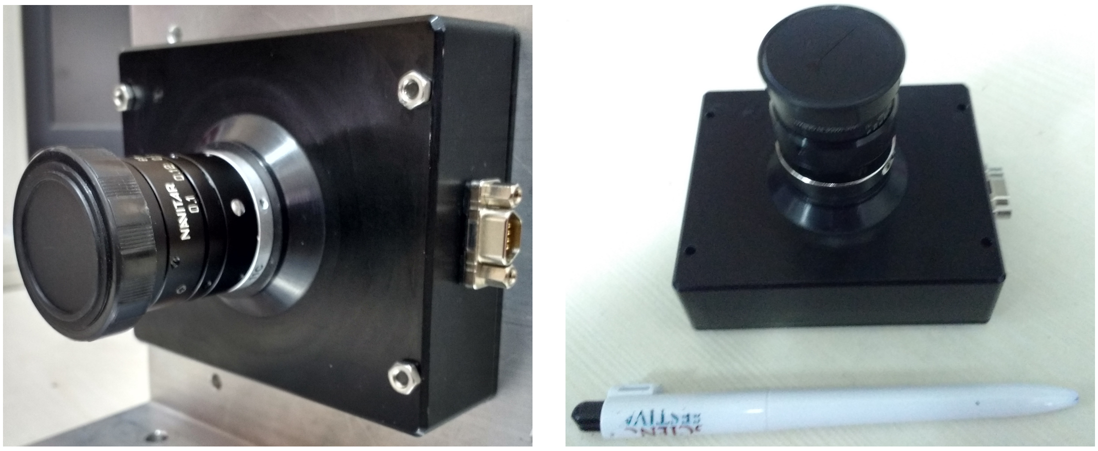


Figure 7.4: Micro star sensor designed for use on cubesats. This is built using many off-the-shelf components deemed suitable for space applications.

stares at a certain field in the sky during control system operation (G. Fritz Benedict , 1998). We can estimate the accuracy of photometry that can be achieved using this *StarSense* camera using detailed photometry analysis.

Furthermore, towards building the balloon observatory platform, the sensor can be integrated with the pointing system in addition with the coarse attitude sensor. The control algorithm for staring observations for the UV payload on the balloon observatory can be developed further using the quaternion outputs of the *StarSense* . The attitude sensor would provide the coarse level pointing accuracy which can be further improved using the *StarSense* as discussed in Chapter 2.

Having scaled the sensor to a cubesat form factor, it can be used on commercial single or constellation cubesat missions. Various attitude control algorithms for these cubesat orientation systems can be developed using the $\mu StarSense$ as the absolute attitude sensor. An integrated system consisting of the sensor, controller and a triplet of reaction wheels as the fine pointing actuator can be developed to be used on cubesats. This would align with the modular design philosophy of cubesats and would be the next step forward.

Bibliography

Sreejith, A. G., Safonova, M., & Murthy, J., “Near ultraviolet spectrograph for balloon platform”, Proceedings of SPIE, 9654, 96540D (2015).

Margarita Safonova, Akshata Nayak, A. G. Sreejith, Joice Mathew, Mayuresh Sarpotdar, S. Ambily, K. Nirmal, Sameer Talnikar, Shripathy Hadigal, Ajin Prakash & Jayant Murthy, ”Overview of High-Altitude Balloon Experiments at the Indian Institute of Astrophysics”, Astronomical and Astrophysical Transactions (AApTr), Vol. 29(3): 397-426 2016.

A. G. Sreejith, Joice Mathew, Mayuresh Sarpotdar, K. Nirmal, S Ambily, Ajin Prakash, Margarita Safonova and Jayant Murthy, ”Measurement of limb radiance and Trace Gases in UV over Tropical region by Balloon-Borne Instruments - Flight Validation and Initial Results”, Atmos. Meas. Tech. Discuss., amt-2016-98, 2016

Hibbitts, C. A., Young, E., Kremic, T. and Landis, R., “Science measurements and instruments for a planetary science stratospheric balloon platform”, *Aerospace Conference, IEEE, Big Sky, MT*, pp. 1-9 (2013).

Navach, C., Lehmann, M., & Huguenin, D., “Feasibility of UV astronomy by balloon-borne observations. I. Stellar spectrophotometry.”, *Astron. Astrophys.*, 22, 361 (1973).

Fesen, R., & Brown, Y., “A method for establishing a long duration, stratospheric platform for astronomical research", *Experimental Astronomy*, **39**, 475, (2015).

Nirmal, K., Sreejith, A. G., Mathew, J., Sarpotdar, M., Suresh, A., Safonova, M., and Murthy, J., "Pointing System for the Balloon-Borne Telescope", Journal of Astronomical Telescopes, Instruments, and Systems, Vol. 02, issue. 04, 047001, (2016).

Ali Siahpush, Janet Gleave, "A brief survey of attitude control systems for small satellites using momentum concepts", Small Satellite conference, AIAA/ Utah State University, (1988).

Guangjun Zhang, "*Star Identification - Methods, Techniques and Algorithms*", Springer (2017).

Spratling, B. B., Mortari, D., "*A Survey on Star Identification Algorithms*", Algorithms, 2, 93-107, (2009)

On semiconductors, "*STAR1000: CMOS Image Sensor, 1 Megapixel, Radiation Tolerant*", Retrieved March 11, 2016, from http://www.onsemi.com/PowerSolutions/product.do?id=STAR1000, (2016)

Erlank, A. O., "*Development of CubeStar A CubeSat-Compatible Star Tracker*", Doctoral dissertation, Stellenbosch University, (2013)

Xilinx, "*Defense-grade Spartan-6Q FPGA Family*", Retrieved March 11, 2016, from http://www.xilinx.com/products/silicon-devices/fpga/spartan-6q.html, (2016)

MATLAB Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States, (2013)

Barjatya, Aroh (2005). Binary search of values in a data vector(http://in.mathworks.com/matlabcentral/fileexchange/7552-binary-search), MATLAB Central File Exchange. Retrieved August 02, 2016, (2005) Kolomenkin, M., Pollak, S., Shimshoni, I. and Lindenbaum, M., 2008, *Geometric voting algorithm for star trackers*, in IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 2, pp. 441-456, (2008).

Markley, F. L., Mortari, D., *How to estimate attitude from vector observations*, Advances in the Astronautical Sciences, 103(PART III), (1979).

Shuster, M. D. 2006, *The QUEST for Better Attitudes*, The Journal of Astronautical Sciences, 54, 657-683, (2006).

Shuster, M. D. and Oh, S., 1981, *Three-Axis Attitude Determination from Vector Observations*, Journal of Guidance and Control, vol. 4, no. 1, (1981).

Wahba, G. 1966, *Problem 65-1, A Least Squares Estimate of Satellite Attitude.* Society for Industrial and Applied Mathematics, 385–386, (1966).

Mayuresh Sarpotdar 2016, *s*tarsense_algorithms, Retrieved August 01 2016, from https://github.com/mayuresh159/starsense_algorithms, (2016)

A. Vyas, M. Roopashree, B.R. Prasad, "Performance of centroiding algorithms at low light level conditions in adaptive optics", International Conference on Advances in Recent Technologies in Communication and Computing 2009. ARTCom'09, pp. 366-369, 2009.

JenaOptronik, " ASTRO 15 Autonomous Star Sensor", Retrieved from https://www.jena-optronik.de/en/aocs/astro15.html, (2015)

Finmeccanica Airborne and space systems division, " Autonomous Star trackers", Retrieved from http://www.leonardocompany.com/documents/63265270/65634713/A_STR_Autonomous_Star_Trackers_LQ_mm07786_.pdf, (2016)

Sodern Ariane Group, "Hydra, A modular star tracker with stunning performance". Retrieved from http://www.sodern.com/website/en/ref/Hydra_316.html, (2017)

Laikin, Milton, "Lens design", CRC Press, (2006).

ON Semiconductors, Phoenix, Arizona, USA.www.onsemi.com."Datasheet: STAR1000 1 Megapixel Radiation Hard CMOS Image Sensor", Rev. 14, Publication Order Number: NOIS1SM1000A/D. January, (2015)

Ramakrishnan, S., Sharma, S.V., Sowmianarayanana, L., "Small Satellite Launch Opportunities on Pslv", Cooperation in Space, Euro-Asian Space Week : where East and West finally meet, Singapore, 23-27 November 1998. Paris: European Space Agency (ESA), ESA-SP Vol. 430, 1999. ISBN: 9290927208, p.547, (1999).

Shahram Mohammadnejad, Ali Bagheri Madineh, Mehdi Nasiri sarvi, "Baffle simulation and design with the aim of use in space cameras", 2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), (2012)

Javad Haghshenas, Ehsan Johari, "A theoretical method for vanes profile design in star sensor baffle", Proceedings Volume 9264, Earth Observing Missions and Sensors: Development, Implementation, and Characterization III; 92641R , (2014)

Hao, Xue-tao, Guang-jun Zhang, and Jie Jiang, "Star sensor model parameter analysis and calibration method", Opto-Electron. Eng 2 5-8, (2005)

Pengju, He, Lian Bin, and Zhang Tao, "Calibration method for wide field of view star sensor." Acta Optica Sinica 31, no. 10: 1023001, (2011)

XIE, Jun-feng, Jian-ya GONG, and Wan-shou JIANG, "An improved on-orbit calibration method for stellar camera [J]." Science of Surveying and Mapping 2: 044, (2009)

Masutti, D., R. Reinhard, F. Singarayar, P. Testani, C. Asma, J. Thoemel, T. Scholz et al, "QB50 System Requirements and Recommendations", Issue 6, Technical report, Von Karman Institue, (2014)

Li, Xuxu, Xinyang Li, and Caixia Wang, "Optimum threshold selection method of centroid computation for Gaussian spot", AOPC 2015: Image

Processing and Analysis, vol. 9675, p. 967517, International Society for Optics and Photonics, (2015).

PHY217, course subject for observational astronomers, “Signal to noise: Photon statistics and CCD equation”,Retrieved from: http://slittlefair.staff.shef.ac.uk/teaching/phy217/lectures/instruments/L14/index.html, (2015)

Fetić, Azra, Davor Jurić, and Dinko Osmanković, “The procedure of a camera calibration using Camera Calibration Toolbox for MATLAB”, In MIPRO, 2012 Proceedings of the 35th International Convention, pp. 1752-1757. IEEE, (2012).

Vaz, Brendon, and B. Eng, “Online calibration for Star Trackers”, Masters’ thesis, Ryerson University (2011).

Mathew, Joice, Ajin Prakash, Mayuresh Sarpotdar, A. G. Sreejith, K. Nirmal, S. Ambily, Margarita Safonova, Jayant Murthy, & Noah Brosch, “Prospect for UV observations from the Moon. II. Instrumental design of an ultraviolet imager LUCI”, Astrophysics and Space Science 362, (2017).

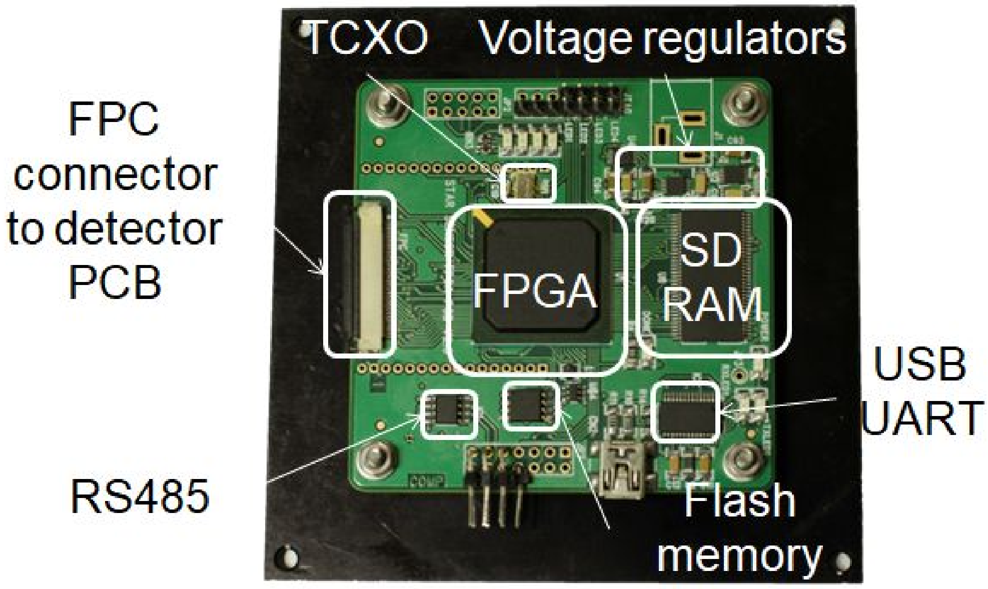
S. Ambily, Joice Mathew, Joice Mathew, Mayuresh Sarpotdar, Jayant Murthy, V. K. Aggarval, S. Nagabhushanam, Divya A. Rao, Sachin Jeeragal, Kaipachery Nirmal, A. G. Sreejith, Binukumar Gopalakrishnan, Margarita Safonova, “PIONS: a CubeSat imager to observe variable UV sources”, Proc. SPIE 10699, Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray, 106993E (2018).

G. Fritz Benedict, Barbara McArthur, E. Nelan, D. Story, A. L. Whipple, P. J. Shelus, W. H. Jefferys, P. D. Hemenway, Otto G. Franz, L. H. Wasserman, “Photometry of Proxima Centauri and Barnard’s Star Using Hubble Space Telescope Fine Guidance Sensor 3: A Search for Periodic Variations*”, The Astronomical Journal, Volume 116, Number 1, July (1998).

Appendix

Appendix - A
StarSense Image Processor Board User guide

Spartan-6Q FPGA Based Readout Card for Star1000 Image Sensor



1. **Introduction:**

StarSense Image Processor board is a real-time image processing FPGA board featuring a MIL-grade Spartan-6 XQ6SLX150T FPGA (150k logic cells). It provides the core functionality of capturing images and processing them in real time for the StarSense to identify stars and calculate quaternions. It implements the readout circuit for image sensor Star 1000 (NOIS1SM1000A) and various real-time image processing algorithms like centroiding, geometric voting and QUEST. It also has FTDI's FT232RL USB-to-UART converter which enables transfer of debug data to a computer. The bit stream for the FPGA is stored on to the SPI Flash memory (N25Q064A 8MB) on board. The flash memory also stores the star catalog required for star identification.

2. **Board Features:**

- FPGA : XQ6SLX150T (in FG484 package)
- DDR RAM: MT46V32M16 166 MHz 512 Mb SDRAM
- Flash Memory: N25Q064A 64Mb SPI Flash memory
- Oscillator : FXO-HC536R-100 100 MHz oscillator
- USB-to-UART Bridge: FT232RL connects to USB
- FPGA configuration via JTAG and Master Serial SPI mode
- On board voltage regulators for single power line operation

How to use the board:

The bit stream for the FPGA is generated in Xilinx ISE. The iMPACT tool from the Xilinx toolsuite sends the configuration bitstream through JTAG. Use this tool we can also program the bit stream on the SPI Flash memory.

We just require a Mini USB cable to connect the board to the computer. Separate external power can be supplied through pin 8 of the IO header on the board.

Master clock:

The FPGA board has one clock source. The 100 MHz oscillator FXO-HC536R which is connected at pin F10. This clock can be scaled internally using a PLL to generate clocks of upto 200 MHz with sufficient jitter characteristics.

GPIO and User LEDs:

The board hosts 4 user LEDs to help debug a code and test fragments of code. It also hosts separate IO pins which can be connected to an external USB controller/ethernet controller to send large image data generated by the FPGA to the computer.

SDRAM interface:

The SDRAM used on board is MT46V32M16 with a maximum operating frequency of 166 MHz. It has been tested at 100 MHz. It works on SSTL 2 logic family which requires Vref = 2.5V. An on board voltage regulator generates the 2.5V required to operate the SDRAM. The SDRAM is connected to bank 1 of FPGA and hence the Vcco for the bank 1 is also equal to 2.5V. The Memory Interface Generator in Xilinx ISE generates a compatible interface for sending user data to the RAM. The RAM interface can also be generated with microblaze in EDK using the axi-s6-ddr IP or s6-mpmc IP. A multi-port memory controller (MPMC) IP is used for this purpose. The MPMC IP can have a maximum of 6 different ports accessing the same memory locations from the DDR memory. A round robin arbitration takes place in prioritizing the memory accesses from different ports on the MPMC IP. We use a native memory interface port to write data to the DDR RAM from the camera capture module along with a PLB bus port to access the data from the MPMC through the microblaze.

Star 1000 interface:

The board was specially designed to interface the image sensor Star 1000 by providing required clocks to readout the image sensor. This interface is implemented using a FPC cable connected to the FPC connector. A 40 contact FPC cable is used for the purpose. This enables developing readout interface for other image sensors too by just replacing the image sensor PCB. The Star 1000 is hosted on a separate PCB. The input-outputs to the Star 1000 can be used with a 3.3V TTL logic level also, hence are connected directly from the FPGA PCB to the image sensor pins. These include the address lines, clock lines, load lines and reset lines. A Verilog IP is designed to generate these signals using a 100MHz clock on board. The outputs of Star 1000 are 3.3V LVCMOS logic levels. This is set by providing 3.3V to Vccio for the image sensor IO bank. The readout of the image sensor is basically sampling data output pins at positive edges of ADC clock and storing these values in the RAM.

Debugging data interface:

The debugging data small in size generated by the FPGA can be sent to the microcontroller on UART (AB17 and Y17) or SPI (W13, W15, U14, Y16). When sending data on UART port, the UART needs to be configured at 9600/230400/921600 baud rate. This debugging interface is through the USB interface provided on the board. In addition to the USB debugging interface, the main interface with the satellite on-board computer is through an I^{2}C interface/ SPI interface or a standard UART interface. An external patch cord made for these interfaces using an FTDI chip would allow connecting to the star sensor board through a computer too.

Satellite interface:

The connectivity to the satellite is provided with an I^{2}C interface where the board acts as a slave on the I^{2}C interface with the On-Board Computer as the master. In addition to an I^{2}C interface, an RS485 interface is also provided to allow standard serial transmissions between the OBC and the FPGA board.

The schematics and the pin connections of the FPGA pins with various other peripherals are as given below. Please read the special notes provided at the end of the schematics before using the StarSense Image Processor Board.

Pin Connections

Peripherals	FPGA Pin
Led1	U16
Led2	Y18
Led3	W17
Led4	AB14
IO0	G9
IO1	G11
IO2	H12
IO3	F15
Mclk	F10
RX_FPGA	AB17
TX_FPGA	Y17

Table 2 Debugging peripherals pin connections

Pin connections Table:

Star 1000 Pin	FPGA Pin	Connector pin
D0	P8	1

D1	R7	2
D2	Y3	3
D3	W4	4
D4	T5	5
D5	T6	6
D6	V3	7
D7	V5	8
D8	P4	9
D9	P5	10
A0	AA1	11
A1	AA2	12
A2	N7	13
A3	N6	14
A4	T4	15
A5	U4	16
A6	P7	17
A7	P6	18
A8	R4	19
A9	T3	20
Ld_y	M4	22
Ld_x	P3	21
S	Y1	26
Reset	M7	25
R	Y2	27
Clk_x	M5	23
Clk_adc	L4	24
G0	W3	33
G1	V1	34
Cal	W1	28

Table 1 Star 1000 Pin connections

Schematics:

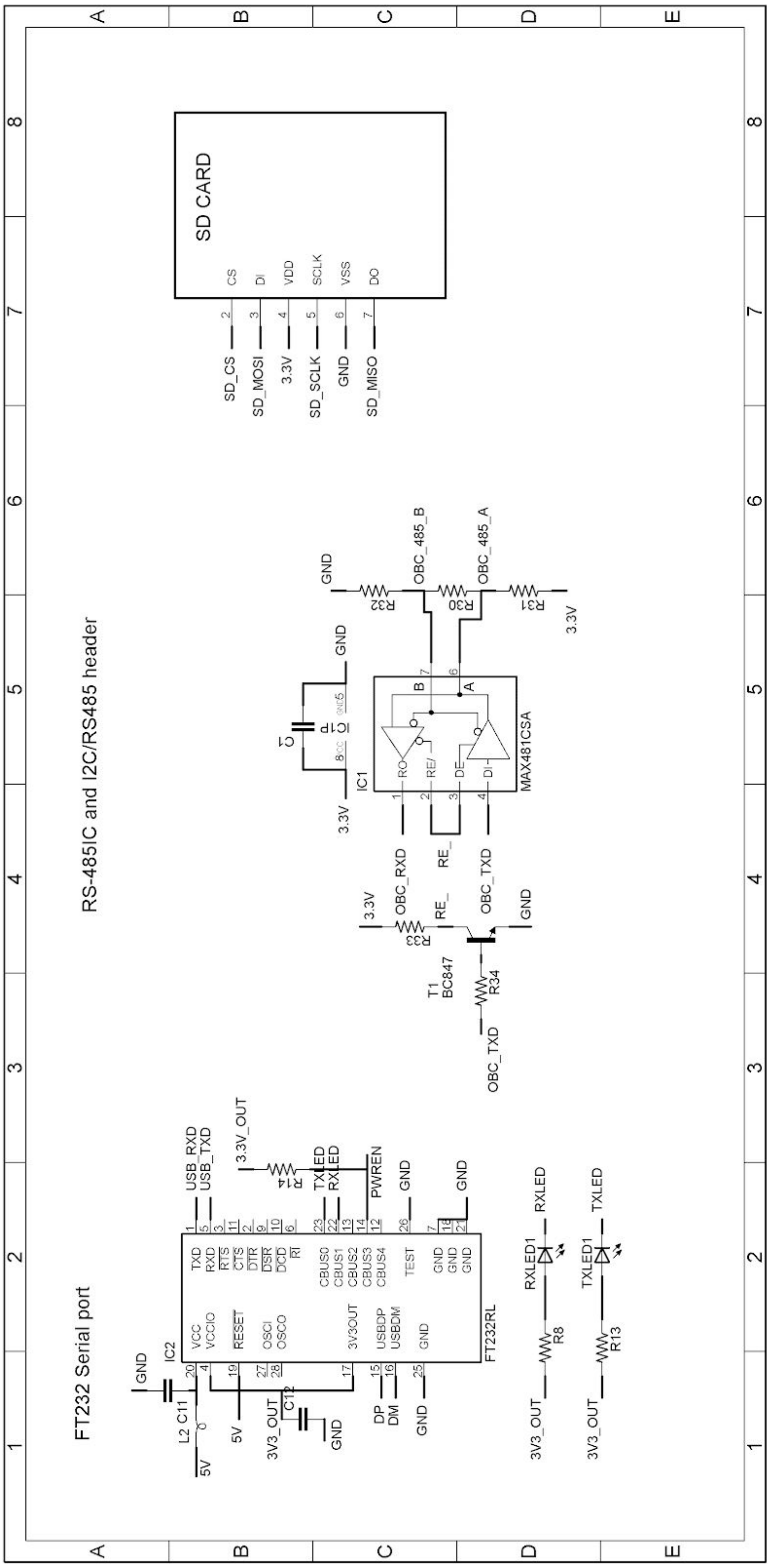


Figure 1 Configuration Microcontroller

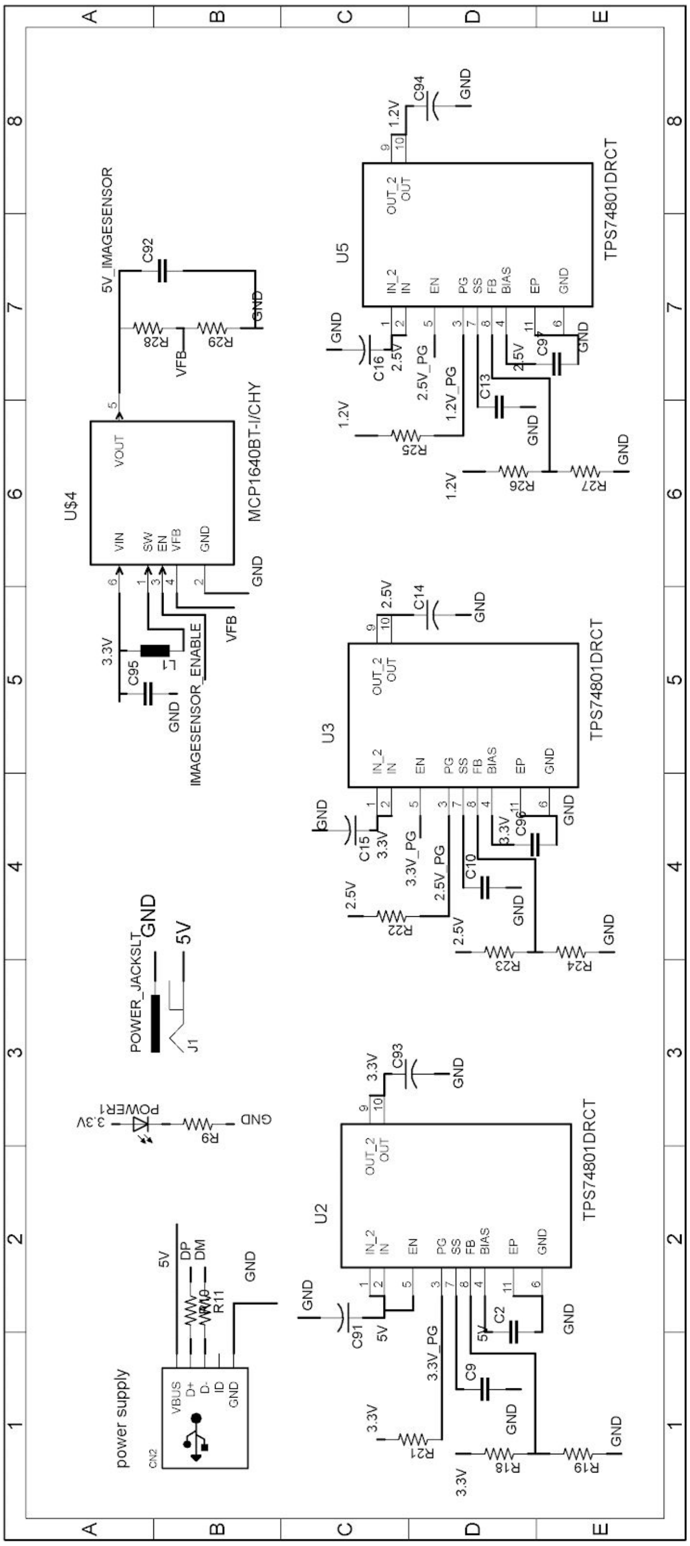


Figure 2 Voltage Regulators

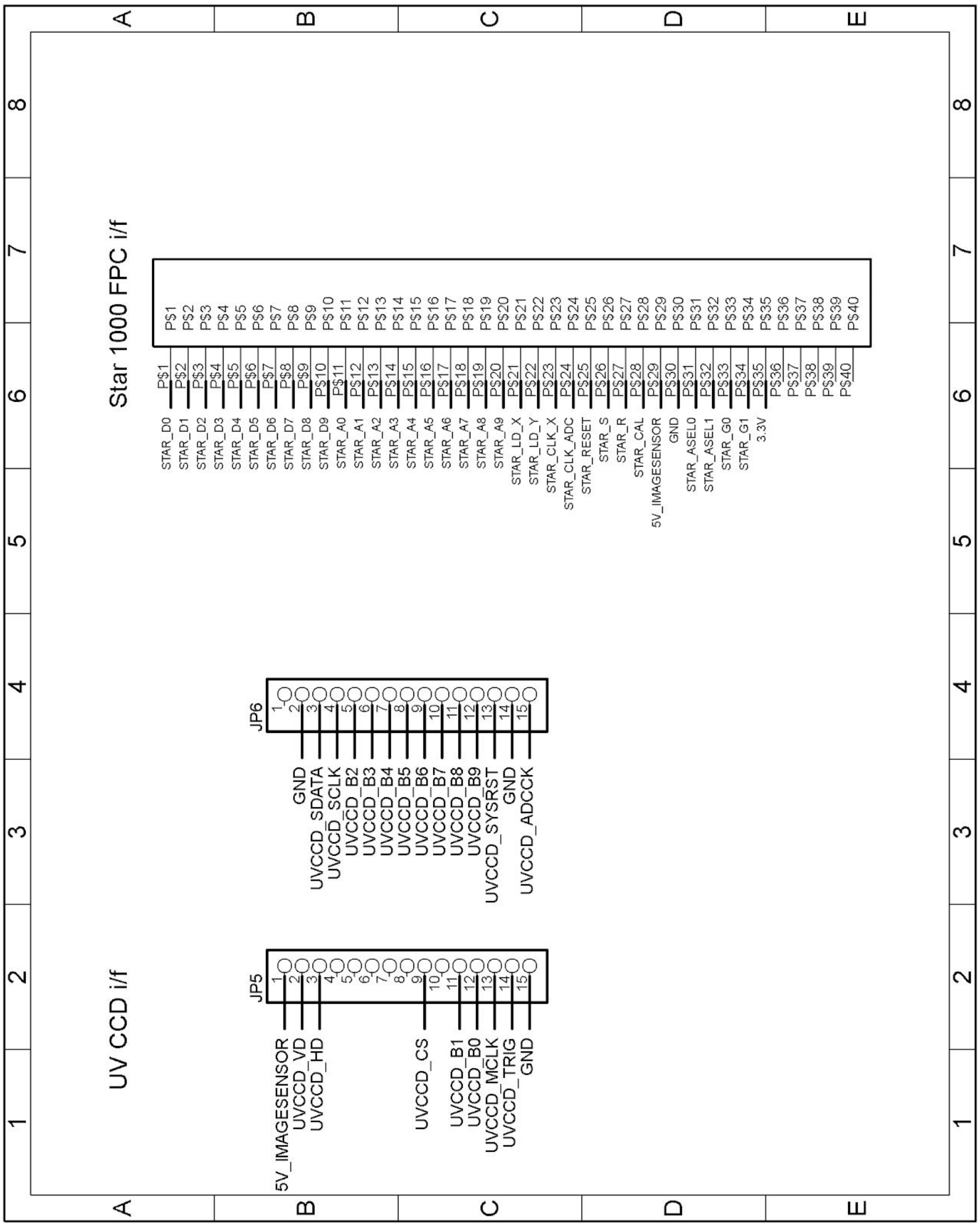


Figure 3 Image Sensor Interface

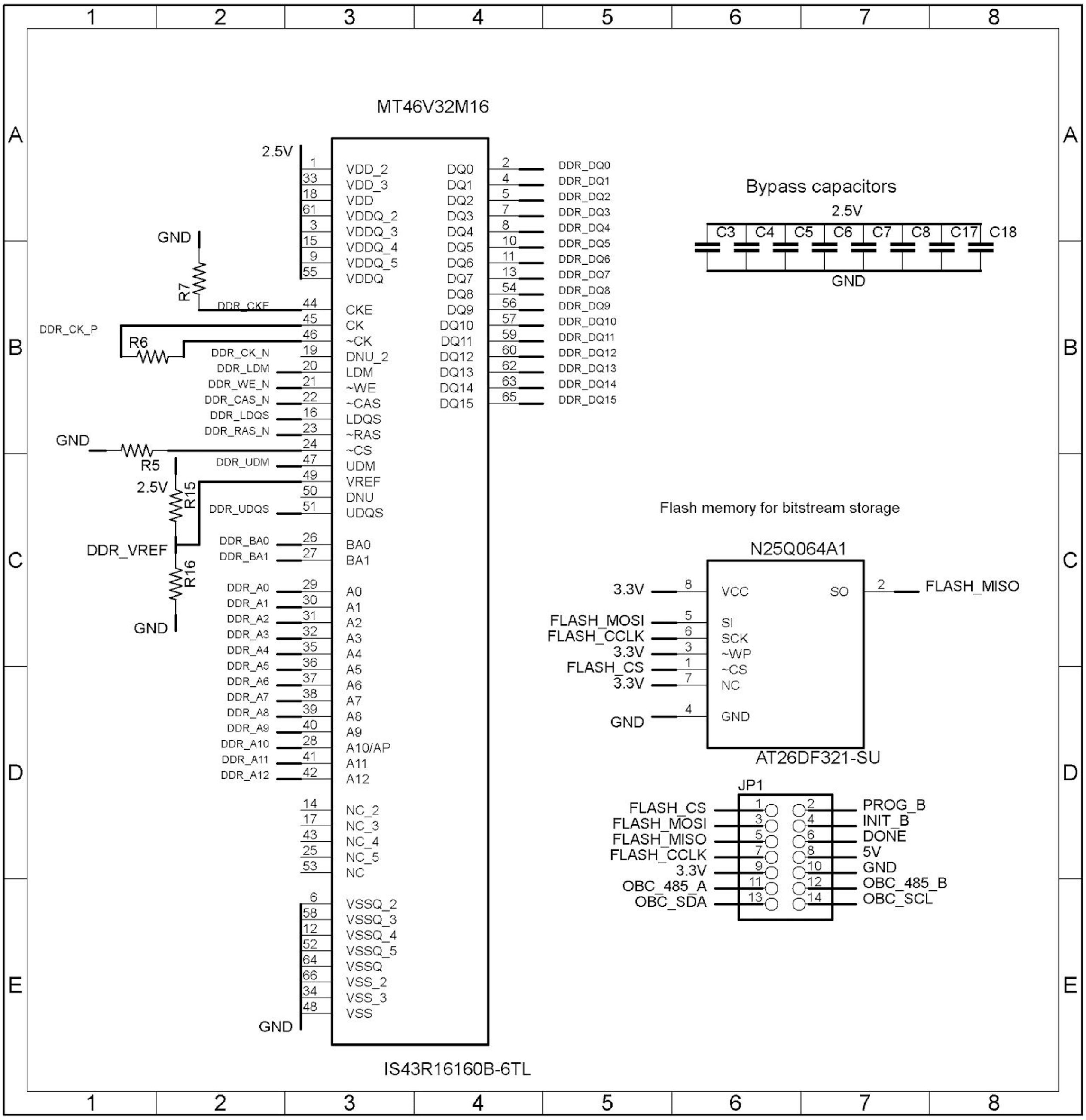


Figure 4 SD RAM connections

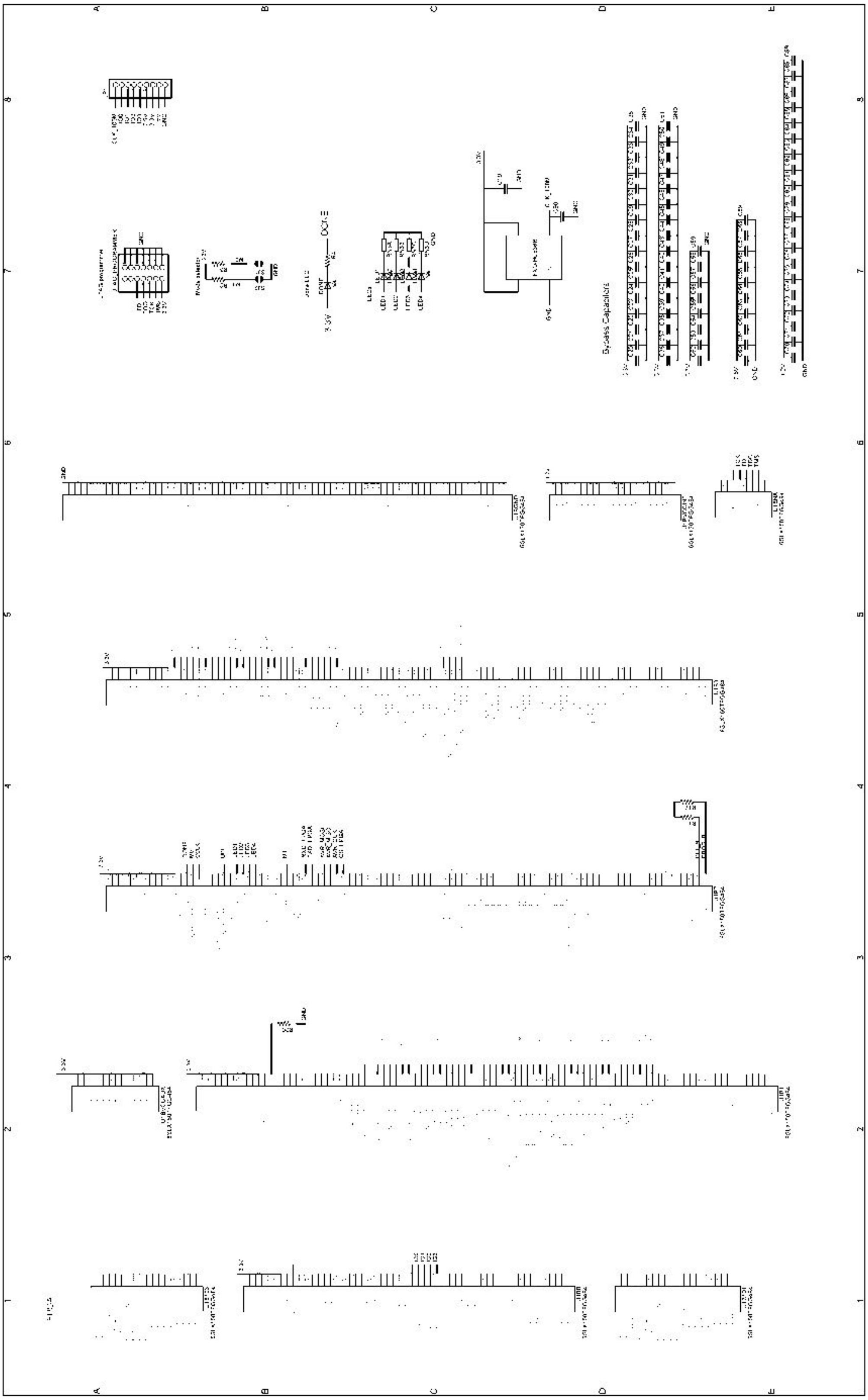


Figure 5 FPGA pin connections

Attention:

- clk_100M pin should not be touched because the oscillator cannot drive any other load other than the HCMOS load of the FPGA pin.
- USB data pins DP and DM are swapped and hence instead of the 22E resistors in series with the pins a cross wire should be soldered to connect the pins properly.

Appendix - B
µ Star sensor

1. **Introduction:**
    a. **Overview:**

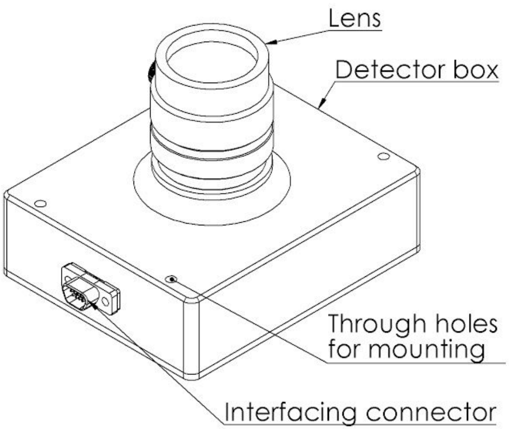
An attitude control system is crucial for payload operations on satellites. Highly accurate sensors which enable the control system to get information about absolute value of current orientation are a critical part in the system. A star sensor is a device which provides accurate information about the current orientation of the spacecraft /satellite with respect to a fixed inertial coordinate system. It is a wide field of view camera which is sensitive to upto stars upto 6^m. The detected stars in the images are identified by pattern matching with a catalog and then optimum quaternion/rotation matrix is calculated knowing the unit vectors of the stars in camera and inertial coordinate systems.

In addition to being a star sensor, the optics of the camera can be used for various scientific imaging applications.

   b. **Highlighted features:**

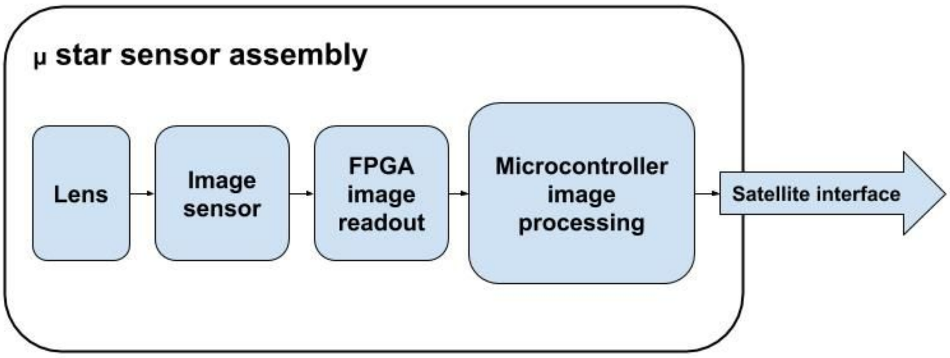
      i. Suitable combination of industrial and MIL grade components used
      ii. Electronics stored in a housing
      iii. Panchromatic imaging camera with wavelength sensitivity from 400 - 1000 nm
      iv. Framing BMP images on board

c. Configuration:



The above diagram shows the configuration of the star sensor. It consists of the lens, the detector box which houses the image sensor and electronics for readout of the image sensor and the on-board image processing algorithms required for the star sensor operation. It can be mounted using the mounting M3 free holes marked in the diagram. The electrical interface between the OBC and the star sensor can be established using the connector shown.

d. Block diagram:



A block diagram of the miniature star sensor is as shown in the above figure. The lens forms the image of the star field on the image sensor. A MIL grade FPGA is used to readout the image sensor and record the pixel values in a RAM. A microcontroller implements all the image processing algorithms on the star field images captured by the imaging setup and calculates the quaternion or the orientation information. This is conveyed to the OBC through the I^2C satellite interface. The power supply for the sensor is also established through the same connector.

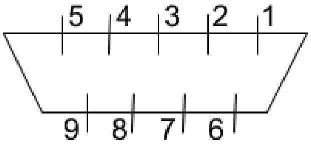
2. Specifications:

- **Dimensions :** 98mmx75mmx69mm (LxBxH)

- **Mass :** 280 gms
- **Power :** 2.3W max
- **Image size :** 1280 x 1024 pixels (~1.25MB)
- **On-board memory :**
  - RAM : 64 MB
  - Flash : 16 MB
- **Operating temperature :** 0^0C to 60^0C
- **Electrical interface :** I^2C interface
- **Pixel scale:** 42”
- **Limiting magnitude:** 5.5^m
- **Catalog size:** ~3000 stars with unit vectors and ~78000 pairs of angles between stars - ~1MB

3. **Data interface:**
   a. **Electrical interface and specifications**
   - Electrical interface : I^2C
     - Single micro D connector for power and data interface with the camera

   b. **Pinout of the connector:**



Micro DB-9 connector on camera - Mating part: **380-009-113L001**

Connector used on the camera side: 380-009-213L001. The datasheet for the connector can be obtained from the following link. The proper mating connector required to connect to this connector is: 380-009-113L001. The datasheet for the same can be obtained from the following link. For the I2C connections, pullup resistors to Vcc=3.3V to be kept on the OBC side. The pinout of the connector is as shown in table.

Pin number	Function	Input/Output
1, 2	+5V	Supply voltage Vdd
3	N.C.	-
4, 5	SCL, I2C clock	Input
6, 7	SDA, I2C data	Inout

8, 9	Gnd	Ground Vss

Commands:

Command	Inference	Response
C	Capture command	R – Capture done, 1 byte
S	Save command	R – Save done as JPEG image, 1 byte
L	List command to return the size of the JPEG file size	4 bytes return for the size of the JPEG file, return value is uint32 datatype
T	Peek command first packet	First 100 bytes of the JPEG image
N	Next packet from the image file	Increment pointer by 100 and send next 100 bytes from the JPEG image
P	Previous packet from the image file	Decrement pointer by 100 and send previous 100 bytes from the JPEG image
I + Int time + Gain	Initialize the image sensor with a certain integration time and gain value for imaging purpose	R - Image sensor initialized
D	Deinitialize the image sensor to put the sensor in low power mode	R - Image sensor initialized

4. **Electrical characteristics:**
   Power dissipation:

Mode	Power dissipation
Switch on (FPGA booting)	0.25W
FPGA programmed (Low power mode)	1.3W
Image sensor readout “ON”	2.3W

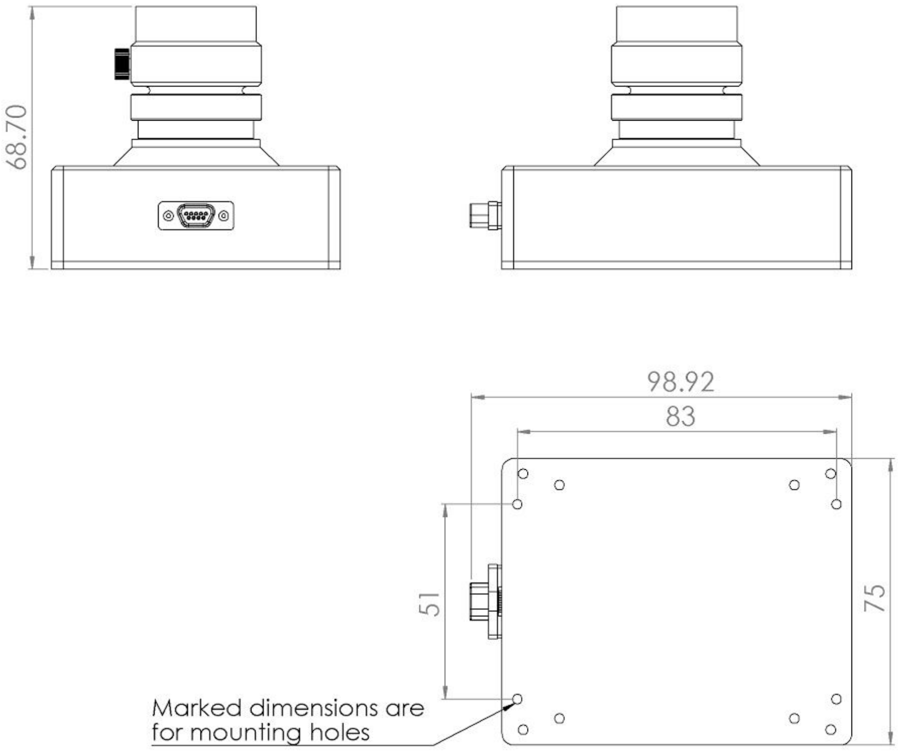
5. **Physical characteristics:**
   a. **Weight table:**

Various components of the sensor used are off-the-shelf components and hence their design specifications are not available. This makes it difficult to be able to predict the center of mass of the sensor very accurately. To be able to crudely estimate the center of mass of the sensor, the weight distribution of the miniature star sensor is as tabulated in the following table.

Component	Mass (gms)
Detector box	108
Lid	36
Lens	88
Image sensor PCB	16
FPGA PCB	29
Connectors	2
PCB Spacers	1
Miscellaneous fixtures	7
Total	287 gms

6. Mechanical drawing:

An outline drawing showing the envelope dimensions and the mounting dimensions of the sensor are shown in the following drawing.

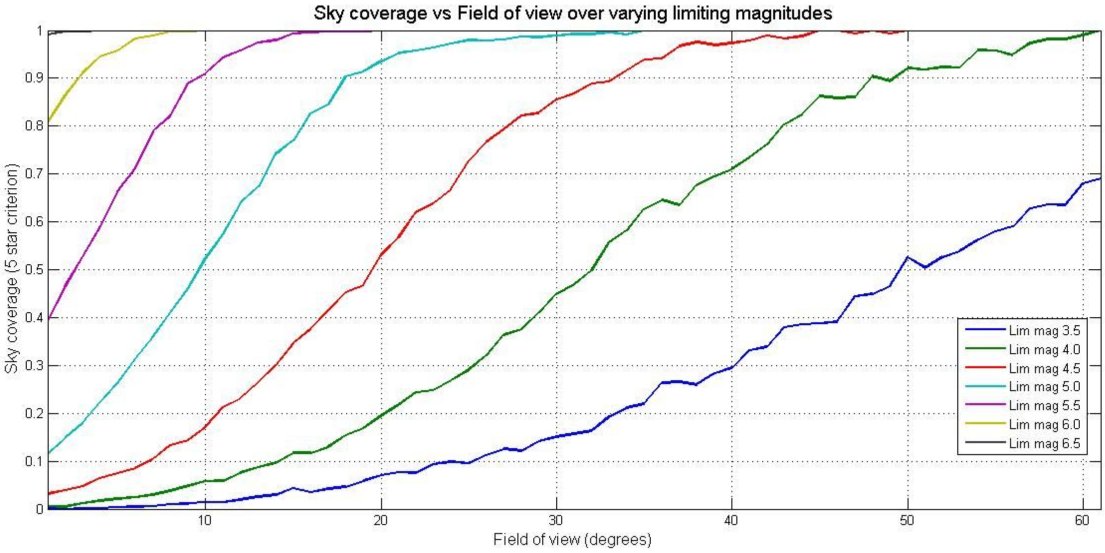

68.70
98.92
83
51
75
Marked dimensions are
for mounting holes

Appendix - C
*μ*Star sensor preliminary test report

1. Introduction:

A star sensor is a very accurate instrument and implements various real-time image processing algorithms.

The design of the instrument is started off with an analysis of the star catalog to be used for the implementation. We selected Hipparcos bright star catalog for use in the star sensor. A sky simulation which obtains the sky coverage of an optical camera with respect to different field of view and varying limiting magnitudes for a criterion of being able to see a minimum of 5 stars in the field was conducted to be able to identify stars and calculate quaternion. The result is as shown in the following figure.



The goal is to obtain high sky coverage to enable the sensor to work at almost any direction in sky. A wider field of view would ensure high sky coverage but at the same time would deter the accuracy of the sensor because of large per pixel angular coverage. A larger limiting magnitude would require larger optics to be able to observe fainter stars which equates to larger diameters corresponding to higher weights. Hence an optimisation is required between the two parameters and the optics to be designed to suit the optimized parameter values. We selected an operating point at 15^0 field of view at a limiting magnitude of about 5.5 which would provide about 95% sky coverage with 5 star criterion.

Following the simulation an optical configuration consisting of an industrial grade, ruggedized, off-the-shelf lens with ~23mm focal length and F/1.4 was selected to be used along with an industrial grade, global shutter image sensor Python 1300 (1280x1024 resolution and 4.8um pixel size). This combination would provide the required field of view (15^0) as concluded from the previous simulation.

The above figure shows the assembled star sensor with the lens, detector inside the box and connector for the interface with a computer. The specifications of the imaging setup are summarized in the table below:

Focal length	22.5mm
F/#	1.4
Image sensor resolution	1280 x 1024
Pixel size	4.8um x 4.8um
Image sensor size	6.144mm x 4.915mm
Field of view	15.54^0 x 12.46^0
Pixel scale	~43"/pixel
Image sensor power dissipation	420mW max

To enable the camera to calculate the outputs various camera parameters have to be passed to the algorithm functions. After having assembled the star sensor with the optics and the electronics in an enclosure, various tests need to be carried out to calibrate the imaging setup. The main tests are as described below:

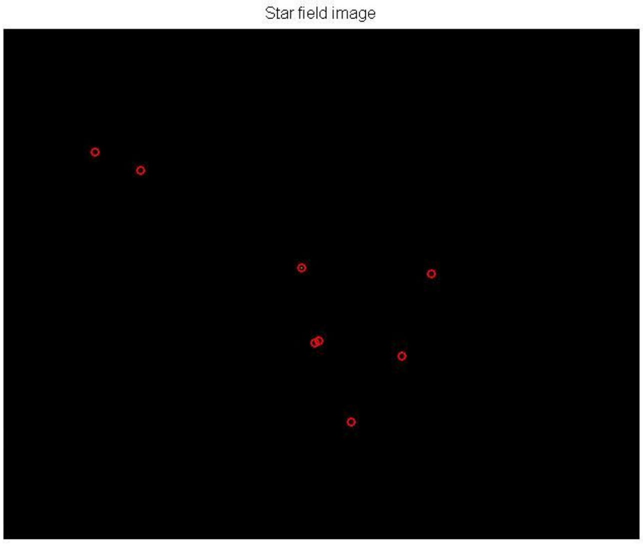
a. Optimize the exposure (gain and integration time) to be able to image stars with the required limiting magnitude.
b. Measuring camera intrinsic parameters like: focal length and principal point.
c. Obtain a model to correct for the distortion introduced by the lens. This model requires distortion parameters to be determined.

2. Star visibility test at Kavalur:

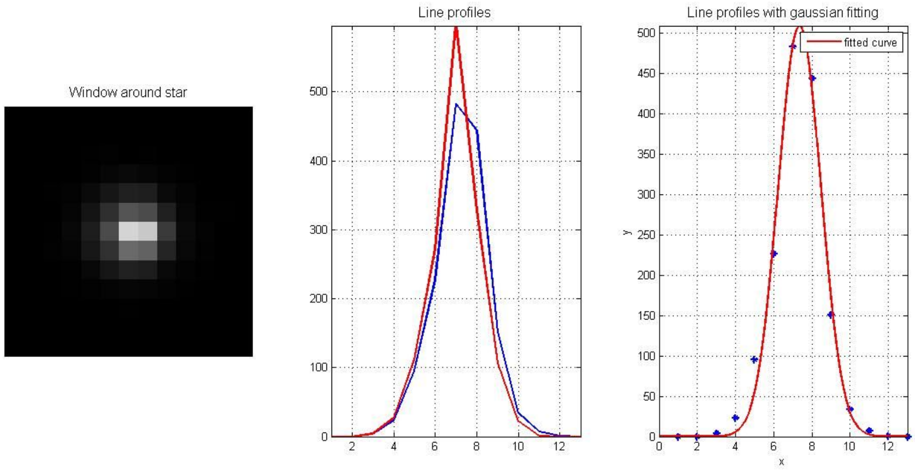
To carry out the optimisation of the exposure, we need to simulate the light intensity coming from the stars. In a laboratory this can be achieved using a collimator and various filters to attenuate the brightness of the source to standard stop values. A more realistic situation is when stars in the sky are directly imaged using the camera. This requires the background sky brightness to be minimum to be able to see the stars with optimal SNR. Having access to such a site which is suitable for astronomical observations, we decided to conduct the test at the Vainu Bappu Observatory at Kavalur, Tamil Nadu. The optimum exposure can be considered as the one where the SNR obtained from aperture photometry for 5.5^m star should be considerable.

a. Gaussian fitting and FWHM testing:

With star field images taken by the camera, we used centroiding function to detect centroids of stars visible in the image. The detected centroids could be obtained as shown below:



Each of the stars detected by the centroiding algorithm are analysed for their Point spread function characteristics. A horizontal and vertical cross cut at the centroid of the star is obtained and a profile is established. A gaussian fitting is used to estimate the FWHM of the spread function. This process is as shown in the next figure.



This gaussian FWHM accounts to about 3 pixels. Thus it can be assumed that effective spread of 3x3 pixels will be achieved for stars on the image plane.

b. Magnitude and photometry:

A theoretical understanding to estimate the required integration time to be able to detect 5.5^m stars is as described further. From standard magnitude reference stars we know the photon flux. After considering all different parameters of the imaging system (viz. Integration time, aperture area, bandwidth, quantum efficiency, pixel efficiency, optics transmission) used to image the stars, number of electrons that can be generated from light collected from a 5.5^m star is obtained. These electrons can be compared with the noise electrons generated in the CMOS image sensor during the integration time and readout of the image.

The signal to noise ratio (SNR) of the photoelectrons vs the noise electrons is obtained and the value is shown to be well above 10. Thus we conclude that the required limiting magnitude can be achieved using the proposed system.

Source characteristics:

Limiting magnitude	5.5
Integration time	30 ms
Focal length of lens	22.5mm
F/#	1.4
Aperture collecting area	2.12 cm^2
Transmission efficiency of lens	0.53 (6 lenses shown in the optical design)
Quantum efficiency	0.5

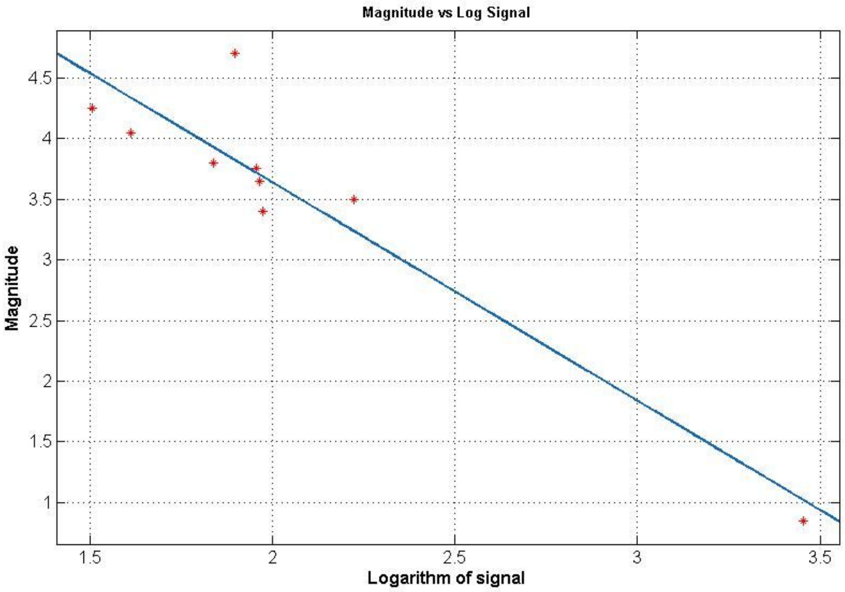
Bandwidth	2500 A
Signal electrons	~350 e

CMOS noise characteristics:

Temporal noise	~7 e
Fixed Pattern noise	~10 e
Dark noise	5 e/s
PRNU	~7 e
Total random noise	~14 e

Thus at a signal from 5.5^m star the Signal to noise ratio obtained in terms of generated photo electrons to CMOS noise electrons is ~25 which is sufficient for successful detection of 5.5^m stars.

In practice we conducted another experiment in terms of ADUs. For different stars with varying brightness, the collected signal is obtained by taking a sum of pixel values over a region of 3x3 pixels as defined by from the spread function. The stars in the image were manually identified by comparing with patterns of stars in a planetarium software stellarium. From the identified stars, their magnitudes were recorded and a logarithm of the total signal is plotted against the magnitude of the corresponding star to establish a magnitude vs signal (in ADUs) relationship for this camera.



$$m = p_1 log(S) + p_2$$

The values for p1 and p2 obtained from the fitting are: (p1 = -1.7, p2 = 7.2).

The signal value would depend upon the integration time and the gain value used in the image sensor. The setting used in this current scenario was integration time = ~200ms and gain = 230.

With these settings the faintest star visible to the centroiding algorithm would be with a signal of about 10 ADUs (selection criterion with 2 pixels at barely threshold value) which would correspond to a magnitude of 5.5^m. Thus almost achieving our required sensitivity.

c. Expected / modelled signal:

The signal in terms of electrons can be easily obtained using the image sensor and optics characteristics known previously. To obtain the expected/model signal value in terms of digital number/ ADU, we need to know the gain (e/DN) of the CMOS image sensor. Signal (in electron units) from different magnitude stars can be obtained using the following equations

$$m = -2.5 log_{10} \frac{F_m}{F_0}$$

$$S_e = F_m * T * A * T_{eff} * QE * \Delta\lambda$$

Where m is the magnitude of the star, F_m is the photon flux at magnitude m, F_0 is the photon flux for reference magnitude (in this case 0th magnitude), T is the integration time, A is the collecting area or the area of the entrance aperture, T_{eff} is the transmission efficiency of the optics, QE is the quantum efficiency of the image sensor pixels (number of electrons generated at every incident photon), $\Delta\lambda$ is the bandwidth of observation of the detector. Furthermore to get the signal in ADU we need to divide S_e by the CMOS pixel gain value.

We conducted a gain calibration test for the CMOS imager. This was done using an integrating sphere with a variable source. The integrating sphere was used to achieve a flat field illumination for the detector and the variation in the experiment was achieved by adjusting the slit width of the source feeding the integration sphere, thus in turn changing the intensity of the input light. The gain (e/DN) can be crudely estimated by using the mean variance characteristics of the images at different illuminations. The experiment was repeated over 4 different digital gain values. The estimation of the gain works because of the following derivation.

$$S_e = gS_c, \ N_e = gN_c$$

S_e, N_e represent signal and noise respectively in electron units, g is the gain value which needs to be determined. We can express total noise as sum of squares of readout noise, photon noise and other flat field noise. But knowing that the arrival of photons from a star in a collecting area is a poisson process and for such a process, signal value is the same as the noise variance (S_e = σ_e^2).

$$N_e^2 = R_e^2 + \sigma_e^2 + \sigma_{0,e}^2 = R_e^2 + S_e + \sigma_{0,e}^2$$

Replacing values in this equation in terms of counts/ADUs

$$N_c^2 = \frac{1}{g}S_c + (R_c^2 + \sigma_{0,e}^2)$$

Thus we know that gain value g is the slope of line representing a graph between signal in ADUs vs noise in ADUs. We determined the mean and variance of signal in each image obtained by varying the source slit width and fitted the corresponding graph with a linear polynomial to get the gain value of the CMOS pixels in the image sensor. After obtaining these CMOS gain values for each of the digital gain settings used in the image sensor, we tabulated the expected signal in ADUs from the previous relation for star magnitudes from 1.0 to 6.0 as follows:

Star Magnitude	Dig Gain = 50, CMOS gain = 3.895	Dig Gain = 100, CMOS gain = 1.915	Dig Gain = 150, CMOS gain = 0.7293	Dig Gain = 200, CMOS gain = 0.8607
1.0	7332	14912	39157	33179
2.0	2919	5936	15589	13209
3.0	1162	2363	6202	5258
4.0	462	940	2470	2093
5.0	184	374	983	833
6.0	73	150	391	331

The values mentioned above are for an integration time of 100ms and analog gain of 2units.

After obtaining this table for a given exposure setting we tested practically in sky condition and compared the obtained ADUs for certain identified stars with those expected at a certain optimum exposure setting (integration time = 100ms, digital gain 160, analog gain = 5.34). 100 images were captured and analysed offline with a centroiding function to detect stars and a window signal calculation function to determine the total signal from a certain star. The following table shows the comparison of the obtained and expected ADUs. We considered the effect of change in analog gain as a multiplication factor to the gain value.

Star	**Magnitude**	**Name**	**Mean DN**	**Expected DN**
1	4.36	HIP 12777	503.84	456.70
2	3.99	HIP 14668	854.41	642.14
3	4.16	HIP 13531	416.05	549.07

4	4.30	HIP 14632	516.28	482.64
5	4.02	HIP 13268	856.64	624.64
6	3.16	HIP 14328	1736.01	1379.21
7	4.59	HIP 16335	501.04	369.51
8	2.00	HIP 15863	3617.21	4014.50
9	4.54	HIP 16826	262.10	386.92
10	5.02	HIP 14382	102.69	248.67
11	3.24	HIP 17358	1206.05	1281.24

d. Noise in signal measurement:

Further we captured images of sky with optimized exposure (integration time = 100ms, digital gain = 160, analog gain = 5.34) to estimate the photon noise from source. As many as 100 images of different parts of sky were captured to measure the signal generated because of each stars and a standard deviation of the signal was obtained to estimate the noise. The result for one such part of the sky are tabulated in the following table. The centroiding algorithm was used to detect stars in the field and match them with a catalog and obtain their magnitudes.

Star	Magnitude	Name	Mean DN	Noise DN
1	4.36	HIP 12777	503.84	148.06
2	3.99	HIP 14668	854.41	124.82
3	4.16	HIP 13531	416.05	125.23
4	4.30	HIP 14632	516.28	69.56
5	4.02	HIP 13268	856.64	222.82
6	3.16	HIP 14328	1736.01	380.17
7	4.59	HIP 16335	501.04	53.28
8	2.00	HIP 15863	3617.21	393.01
9	4.54	HIP 16826	262.10	26.13
10	5.02	HIP 14382	102.69	84.97

11	3.24	HIP 17358	1206.05	160.42

3. Camera intrinsic parameters and distortion coefficients measured using checkerboard charts:

The intrinsic parameters of the camera are used in calculating the unit vectors of detected stars in camera coordinate system which are in turn used to calculate angle between detected stars and also to calculate the final quaternion while using the QUEST algorithm. Thus, the accuracy with which these parameters are identified is very crucial for accurate quaternion measurements.

For the current star sensor, we measured the intrinsic parameters using checkerboard images and a MATLAB tool to calibrate the camera. We got a checkerboard printed on a A0 size chart paper and placed it approximately at the hyperfocal distance of the camera. (Hyperfocal distance is the distance space beyond which is infinity for the camera and comes to focus at the focal plane of the lens). For the focal length of the lens currently being used in the camera ~10m is the hyperfocal distance. We placed the checkerboard chart about 15m away from the camera and captured about 10 images at different locations of checkerboard on the image plane. A sample image is as shown in the below figure. These images are input to the MATLAB camera calibration program along with the physical size of the squares being used. Output of the toolbox is the camera calibration parameters.

A radial distortion model was fitted in the vertices of the checkerboard extracted by the image processing algorithm in the toolbox. The coefficients required for correction of the radial distortion introduced by the lens were also obtained from the MATLAB toolbox. The equation used for correction of the distorted image is as follows:

$$x_u = x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4)$$
$$y_u = y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4)$$
$$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$$

In runtime of the star sensor, this correction is applied after the centroids of the stars are calculated to avoid spending excess time for the complete image distortion correction.

The camera parameters and the distortion coefficients are obtained after optimisation such that the reprojection or residual errors remaining in the image are of the order of 0.1 pixels.

The parameters obtained are as noted below:

Focal length: 22.36mm. (Specification of the lens - 22.5mm)

Principal point: 355.75, 343.02

Distortion parameters: (-0.2456, 0.1668)

Sample checkerboard image

After the above mentioned processing, the Right Ascension and Declination value of the principal point is obtained using certain formulae with quaternions. This is done offline to evaluate the images as well as quaternions.

A more detailed testing is in plan for live centroiding, star identification and quaternion calculation to obtain quaternions at required rate of about 5 Hz.