INSTITUTE OF CONTROL AND COMPUTATION ENGINEERING
FACULTY OF ELECTRONICS AND INFORMATION TECHNOLOGY
WARSAW UNIVERSITY OF TECHNOLOGY

MASTER OF SCIENCE THESIS

STAR-TRACKER PROGRAM FOR CUBESAT SATELLITES

Szymon MICHALSKI

Supervisor:
prof. dr hab. inż. Ryszard Romaniuk

Warszawa 2017

Abstract

Last years directed space industry towards small satellites. Many countries which did not have possibility to enter this branch of industry before now create their own solutions. The goal of this work is to create fully functioning star-tracker software eligible to be used in future satellites as Polish solution of determination of satellite attitude towards Earth. Work contains also description of individual parts and variants of solutions connected with star-tracker.

Streszczenie

Ostatnie lata ukierunkowały przemysł kosmiczny na małe satelity. Wiele krajów, które wcześniej nie miały możliwości wejścia w tę gałąź przemysłu, teraz tworzą własne rozwiązania. Celem niniejszej pracy dyplomowej jest stworzenie w pełni działającego oprogramowania star-trackera nadającego się do wykorzystania w przyszłych satelitach jako polskie rozwiązanie problemu określania orientacji satelity względem ziemi. Praca zawiera też opis poszczególnych części i wariantów rozwiazania problemów związanych ze star-trackerem.

Contents

Acronyms

ADCS Attitude Determination and Control System.

GPS Global Positioning System.

LEO Low Earth Orbit.

LIS Lost-In-Space.

List of Symbols

ϕ Euler angle, roll.

θ Euler angle, pitch.

ψ Euler angle, yaw.

$\boldsymbol{R}(\cdot)$ Rotation matrix using Euler angles.

$\boldsymbol{q}$ Unit quaternion.

q_0 Scalar part of unit quaternion.

$\boldsymbol{q}_{vec}$ Vector part of unit quaternion.

$\boldsymbol{Q}$ Quaternion matrix.

v General Euler angle.

$\boldsymbol{n}$ Unit vector.

$\boldsymbol{I}$ Identity matrix.

$\boldsymbol{S}(\cdot)$ Skew symmetric matrix.

$\boldsymbol{R}_n^b$ Rotation matrix representing a rotation from n to b.

$\boldsymbol{M}$ Least squares estimate of rotation matrix.

$\boldsymbol{r}$ Known directional unit vector in the NED frame.

$\boldsymbol{b}$ Known directional unit vector in the BODY frame.

1 Introduction

Stars were used for navigation already ages ago. Together with technological development, spread of sailing and seafaring people started to use more advanced tools helping to more precisely estimate position of ships on sea. In XVIII century new tool was created, named sextant. This device helped to quite precisely calculate angle between the line ship-horizon and ship-star, what let to estimate position.

Nowadays, hundreds of years later, thanks to other technologies like for example Global Positioning System (GPS) stars are not necessary any more for travelling. As it is commonly known, GPS technology is based on satellites, and today those satellites need stars for determining their attitude towards Earth.

With the miniaturization of electronics and batteries new type of satellites was invented: CubeSat micro-satellite. They have now greater sensory and processing power possibilities, previously found only in larger satellites. However CubSats are still behind in terms of attitude determination.

1.1 Motivation

The goal of this work is to make fully operational star-tracker software, that could be used on Cubesat satellites. Such program could be used on space missions and could start Polish state-of-the-art technology in growing space technology sector. There is already existing prototype of on-board computer for such CubeSat satellite, which together with this work should create full star-tracker device: hardware + software.

1.2 Outline of thesis

This thesis consists of several chapters. Here they are shortly summarized:

Chapter 1 serves as introduction to this thesis and describes the motivation and goal of this work. It also describes the background of the topic.

Chapter 2 describes all the important foundations for the fully understanding given work.

Chapter 3 is the main part of this thesis. It describes how the star-tracker program works and goes through detailed comparison of different approaches.

Chapter 4 describes the created prototype of star-tracker in Python language.

Chapter 5 talks about the implementation of star-tracker on the existing prototype of on-board computer.

Chapter 6 describes how the finished program is performing.

Chapter 7 contains conclusions about this work and created star-tracker program.

1.3 Related work

There exists a number of works connected to topic of star-tracker. Many works are cited later in this thesis as they describe important parts of algorithms. There are just a few works dealing with whole start-tracker program, not just parts of it, and nearly no works about whole program together with implementation on the real equipment. The related work can be divided into few sections:

- Works describing CubeSats in general: Swartwout [1] and Heidt et al. [2].
- Algorithms for calculating star centroids: Liebe [3], Samaan et al. [4], Knutson [5], Azizabadi et al. [6], Lindh [7] and Zhang et al. [8].
- Star identification (which star in image corresponds to which star in on-board catalogue) divided by design:
  - Planar Triangle - Cole and Crassidis [12],
  - Spherical Triangle - Cole and Crassidus [14],
  - Pyramid - Mortari et al. [13],
  - Geometric Voting - Kolomenkin et al. [9],
  - Grid Algorithm - Padgett and Kreutz-Delgado [10],
  - Brightness Independent - Dong et al. [11],

  - SP-Search - Mortari [15],
  - use of neural networks:
    * Miri and Shiri [16],
    * Lindblad et al. [17],
    * Li et al. [18],
  - separately discussed k-vector algorithm for faster search:
    * Mortari [19],
    * Mortari and Neta [20],
    * Mortari and Rogers [21].
- Attitude Determination divided by design:
  - AIM (Attitude estimation using Image Matching) - Delabie [24],
  - QUEST - Shuster [25],
  - QUEST improvement - Cheng and Shuster [26],
  - Extended Quest - Psiaki [27],
  - EQUEST - Rinnan [28],
  - Singular Value Decomposition - Juang et al. [29],
  - Optimal Image Matching - Delabie [24]
  - summarized description of attitude estimation algorithms:
    * Markley and Mortari [30],
    * Hall [31],
    * Tappe [22].
- Works on hardware for star-tracker Azizabadi et al. [6], Gąska [32], Felikson et al. [33].
- Works on full star tracker but without implementing on real device/simulations only: Kandiyil [34], Huffman et al. [35] and Diaz [36].
- Full star-tracker with device/on real device Jalabert et al. [37], Lizy-Destrez and Mimoun [38], Rose [39], Mortari and Romoli [40] Cannata et al. [41].

Many of those works will be mentioned and cited later in this thesis as they are crucial for this work and describe the ways how star-tracker should be designed. All those main algorithm types, like finding star centroids or star identification, are also described here in full detail in next sections.

1.4 Cubesat

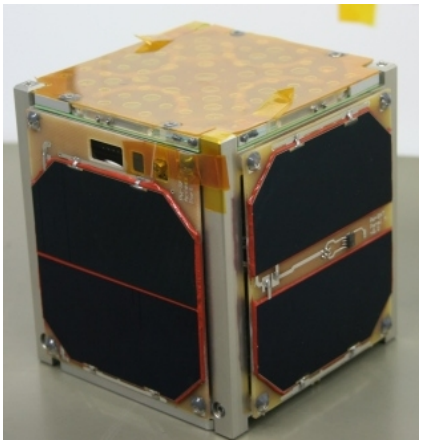
In recent years there has been the development of micro-satellites called CubeSats. CubeSat is a standard created in 1999 at the California Polytechnic State University [2], which is used for low-cost micro-satellites. CubeSats are measured in units. Most are 1U, 2U and 3U. CubeSat 1U has a size of 10 cm on each edge and the maximum weight of 1.333 kg, while the CubeSat 2U is 20x10x10 cm and can weigh up to 2.666 kg.

The advantage of the standard is primarily reduction of the price of elevation of such satellites into orbit. Their popularity is evidenced by the fact that the percentage of satellites weighing less than 10 kg has increased to about 60% of all satellites, and only CubeSats make up about half of the small satellites that were launched in last years [1][42]. Till now there were around 800 CubeSats launched [43].

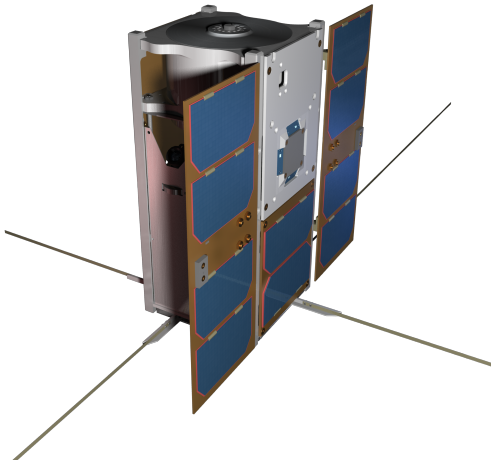
There exist wide range of CubeSat examples. Mostly they are scientific and students' satellites, for learning, experimental and science purposes. There are also few Polish examples: PW-SAT (1U) and PW-SAT 2 (2U) designed by students of Warsaw University of Technology, and Lem (8U) and Heweliusz (8U), both made at Polish Academy of Sciences. PW-SAT and PW-SAT 2 were used for learning and experimenting with new technologies, like deorbitation sail, solar sensor, etc. [44]. Scientific satellites Lem and Heweliusz were built as part of BRITE programme - joint programme of Austria, Canada and Poland. The goal of this programme is to research mechanisms of energy transportation and angular momentum which happen inside hottest stars [45].

Typical CubeSat consist of many different components. Quite often it has antennas and radios for communication with mission centre on Earth, Electric Power System (EPS - subsystem responsible for managing electric power in the satellite), Attitude Determination and Control (ADCS - subsystem responsible for analysing data from sensors and taking decision about trajectory, etc.), magnetometer, star-tracker, sun sensors, payload processor with mission instructions, etc. CubeSats are quite small, hence they usually do not have any propulsion system. Figure 2 shows the components of CubeSat on example of NASA's Interplanetary NanoSpacecraft Pathfinder In a Relevant Environment (INSPIRE).

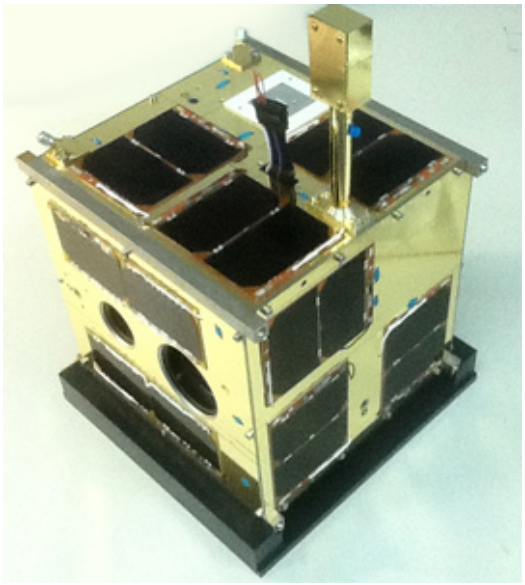
Costs of such CubeSat satellite on example of PW-SAT 2: 120,000-



(a) PW-SAT [46]



(b) PW-SAT 2 [44]



(c) BRITE-PL (Lem, Heweliusz) [47]

Figure 1: Example of Polish CubeSats.

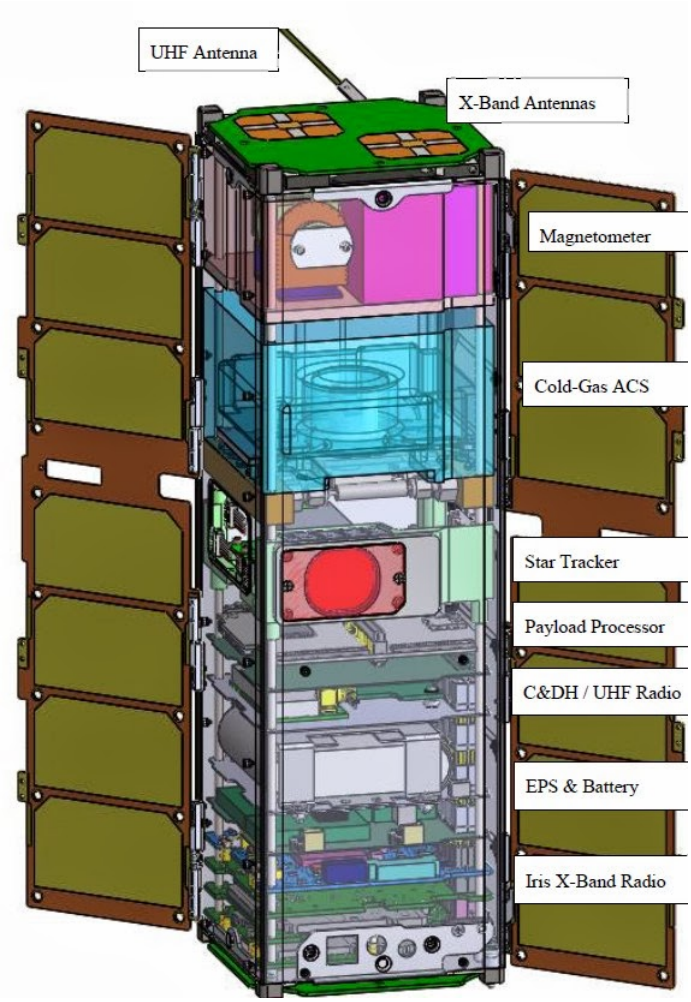


Figure 2: CubeSat build - INSPIRE, Image [48]

200,000 EUR for putting the satellite on the orbit and round 70,000 EUR for designing and building the satellite itself [49].

1.5 Means of attitude determination

Attitude of spaceships must be usually stabilised and controlled for various reasons. It is necessary for satellite antenna to be pointing towards Earth for the proper communication, to intelligently control the heat by using the effects of cooling and heating of the shadows and sunlight, as well as to navigate: manoeuvres must be performed in the right direction.

Attitude is determined between two coordinate systems (where one is reference system) and defines by what angles the coordinate system connected with the researched object has to be shifted in order to cover the reference system. Devices such as planes and satellites have so called Attitude Determination and Control System (ADCS), which controls attitude of object relative to an inertial reference frame or another entity (the celestial sphere, certain areas, the nearby objects, etc.).

Currently, attitude determination of CubeSats is limited mainly to the sun sensors, magnetometers and measurements of inertia. The following table describes the accuracy of different sensors. Unquestionable winner here is the star-tracker, which, due to its quality and uniqueness of the constellations is ideal for navigation.

1.5.1 Inertial Measurement Unit

An Inertial Measurement Unit (IMU) detects linear acceleration using one or more accelerometers and rotational rate using gyroscopes, sometimes they also include magnetometers. They are commonly used in planes, spacecrafts and many consumer devices like smart-phones. The biggest disadvantage of IMU is accumulated error. Even small error that is accumulated with time brings device to think it is in different location than it actually is. Usually the errors are corrected via use of Kalman filter, which corrects IMU errors using other sources (i.e. GPS).

1.5.2 Sun Sensors

Sun sensors typically consist of few sensors oriented perpendicularly, which can tell the angle of the sun. They can give the information how the satellite is oriented towards the sun.

1.5.3 Star-Tracker

Star-tracker is quite sophisticated device, which consists of camera, processor and memory for database. In the database is catalogue of stars, built basing on data from Earth's observatories. Simply put star-tracker, when in orbit, takes picture of space - stars. Then processes the image to recognize stars in it and compares with data in catalogue or with previous image. When stars are identified star-tracker estimates the attitude of satellite towards Earth. The main advantage over other means: great accuracy and star-tracker can find its location again after satellite was restarted or lost in space.

1.5.4 Horizon Sensors

Earth horizon sensors use infra-red radiation from the Earth's surface via use of bolometer. The device detects when infra-red signal was first detected and then lost. The time between is used to determine the Earth's width, what later can be used to determine roll angle. In other words device detects the contrast between the cold of deep space and the heat of Earth's atmosphere. Problems of this sensors are the fact that Earth is not perfectly circular and sensors detect infra-red in the atmosphere, which varies depending on the latitude.

1.5.5 Magnetometer

The device actually consists of three magnetometers - one for measuring Earth's magnetic field in each direction. Measurements can be compared with known magnetic field given by International Geomagnetic Reference Field model. The problem with this sensors is that they are susceptible to noise and outside of Earth's orbit it is completely unpredictable.

1.5.6 Global Navigation Satellite System

Nowadays nearly everyone has personal device with Global Navigation Satellite System (GNSS). They are more commonly known by their implementation by various countries: Global Positioning System (GPS) by USA, GLONASS by Russia, Galileo by European Union, and in close future BeiDou by China. It is possible to use GNSS to determine spacecraft's attitude. This is possible when at least four GNSS satellites are visible by spacecraft, what provides three-dimensional position. Even though it gives quite good accuracy, but it decreases with the altitude. GPS satellites are located on Medium Earth Orbit (MEO), around 20,200 km above sea [50]. It is possible to navigate spacecraft between Low Earth orbit (LEO - 160 to 2,000 km) and Geosynchronous Orbit (GEO - 35,786 km) [51]. For missions beyond Earth's orbit GNSS navigation is however not useful. Example of GNSS usage for navigation in space is International Space Station (ISS), which uses GPS. [52]. There are however works that claim it could make using GPS systems possible for attitude determination up to Moon attitude in the future [53].

1.5.7 Conclusions

The above means of attitude determination are usually used together with the use of Kalman filter, to compensate errors of one sensor with the other. However none of the means is more accurate than star-trackers. Sun sensors can provide very accurate measurements, but can only operate in sunlight. For low-Earth orbit (LEO), even 30% of the orbit can be done in the darkness. Magnetometers are small and can give accurate measurements if properly calibrated. Their drawback is the limited knowledge of the magnetic field and electromagnetic interference due to highly integrated construction of CubeSats. Microelectromechanical gyroscopes are small enough to fit into a CubeSats. However, they suffer from sudden movements, and could not maintain the correct measurement of the 15-minute period of the eclipse orbit LEO. GPS navigation is accurate, however only on Earth's orbit. To be truly competitive and reliable platform, CubeSats must provide the correct determination of attitude. The best way to meet this goal is via using star-trackers. The typical accuracies of different sensors are available in Table 1.

Sensor	**Accuracy** (less is better)
Inertial Measurement Unit	0.001°/hr to 1°/hr
Sun Sensors	0.005° to 3°
Star Sensors	0.0003° to 0.01° (1 arc sec to 1 arc min)
Horizon Sensors	0.05° (GEO) 0.1° (LEO)
Magnetometer	1.0° (5000km alt) 5.0° (200 km alt)

Table 1: Sensor Accuracy Ranges. Adapted from Hall [31] and Larson and Wertz [54]

1.6 On-board computer

This section will describe the on-board computer which was done as part of other thesis.

2 Preliminaries

2.1 Earth's orbits

Most of spacecrafts' missions take place on Earth's orbits. There exist a variety of different classifications of orbits. Here will be presented only a few necessary to understand the topic.

Centric classification:

- geocentric - orbit around Earth
- heliocentric - orbit around the Sun
- Lunar orbit - orbit around the Earth's moon

Orbital period classification:

- geosynchronous - period of rotation is equal to one sidereal day (23 hours, 56 minutes, and 4 seconds) in the same direction as Earth. This results that satellite stays in the same meridian, but can move in the North-South axis. The altitude of such orbits is 35,786 km above sea level.
- geostationary - Special case of geosynchronous orbit. A geostationary orbit stays exactly above the equator. For the observer on the surface the satellite stays always at the same point in the sky. Most commercial communication, broadcast and Satellite-Based Augmentation System (SBAS - system complementing GNSS) satellites use geostationary orbits.
- semi-synchronous - orbit has an orbital period of ½ sidereal day. The example here are orbits of GPS satellites.

Altitude classification:

- Low Earth orbit (LEO): altitudes from 160 to 2,000 km. Used among others for ISS, Earth observation and spy satellites, some communication satellites like Iridium phones network, Hubble Space Telescope. Every object on altitude below 160 km will quickly loose it's altitude and it's orbit will decay.
- Medium Earth orbit (MEO): altitudes from 2,000 km to 35,786 km. Used for navigation, communication, space environment science. Most commonly used is the altitude approximately 20,00 km, because it has orbital period of ½ sidereal day. It is used for example by GPS. Other GNSS satellites' orbits are 19,000 km for GLONASS and 23,222 km for Galileo.
- Geosynchronous (GSO) and Geostationary (GEO) orbits - altitude approximately 35,786 km.
- High Earth orbit: altitude above 35,786 km. Orbital periods are longer than 1 sidereal day.

The mentioned orbits are visible to some extend in the Figure 3. Of course this is just a small variety of possible orbits, however describing more is not necessary for the purpose of this work.

2.2 Coordinate frames

Describing attitude is not that simple outside Earth's surface. Earth rotates and moves around the Sun, therefore it is necessary to understand means needed for correct attitude description. Below are described a few important frames important for understanding this work.

2.2.1 ECEF frame

Earth-Centered, Earth-Fixed frame is visible in Figure 4. Its x axis points towards intersection between Greenwich Meridian and equator, while its z-axis points along the rotation axis of the Earth. The y-axis completes a right handed orthogonal coordinate system. The origin of the frame is at the center of the Earth. This all means that the coordinates move together

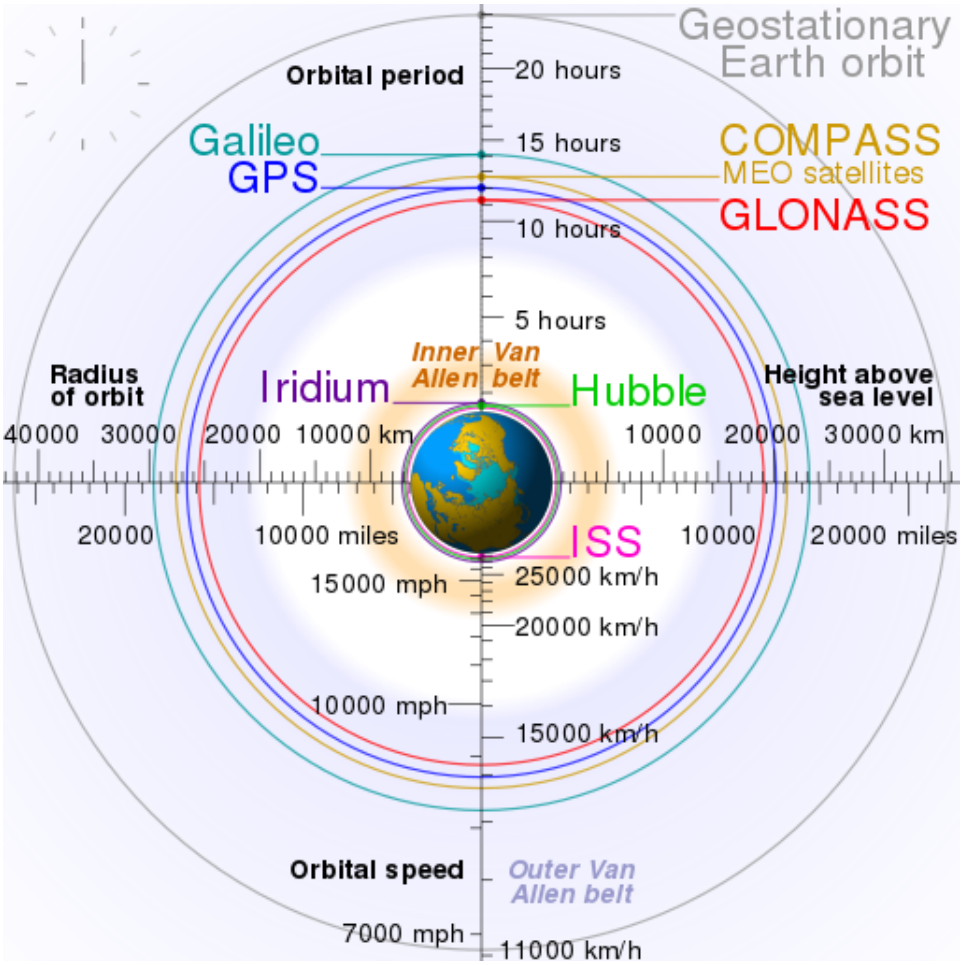


Figure 3: Earth's satellites navigation orbits [55]

with Earth's rotation. It is good frame for representing objects on Earth's surface, but not for objects in space.

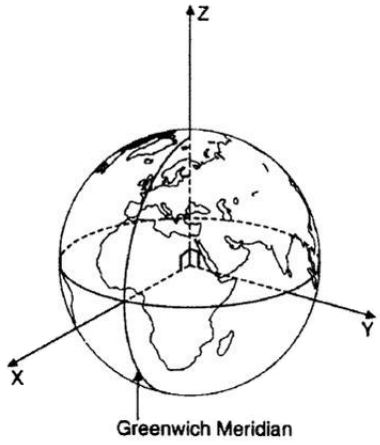


Figure 4: ECEF frame, Image from Larson and Wertz [54]

2.2.2 ECI frame

The Earth Centered Inertial frame is similar to ECEF, with the difference that it is inertial - it does not follow Earth's rotation, but stays the same despite the planet's rotation. The origin of the frame is at the center of the Earth. It is much better frame for representing objects in space. Larson and Wertz [54]

2.2.3 NED frame

The North East Down frame is represented in Figure 5. Its z-axis points downwards, perpendicular to the tangent plane of Earth. The x-axis points towards true north and the y-axis points East. The NED frame is an inertial frame - not dependent on Earth's rotation.

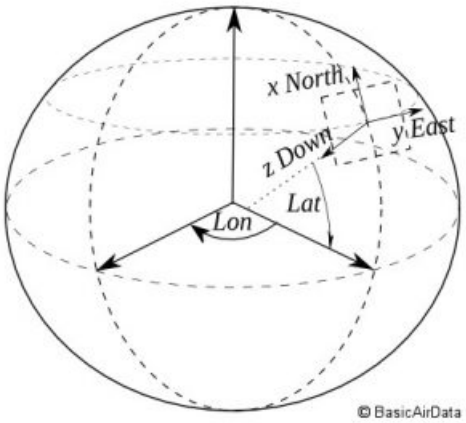


Figure 5: NED frame [56]

2.2.4 BODY frame

In this frame the origin is the centre of the spacecraft. The moves and rotates with the spacecraft. The x-axis points forward from spacecraft, the y-axis points to the right side and the z-axis points downwards. The frame is represented in Figure 6.

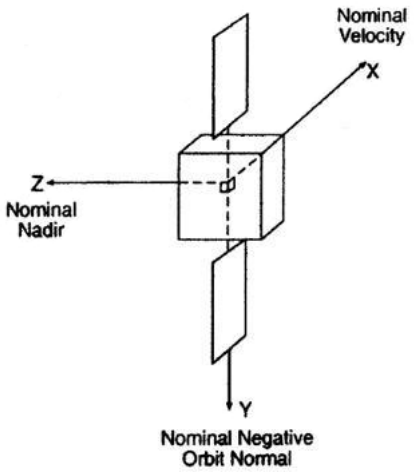


Figure 6: BODY frame, Image from Larson and Wertz [54]

2.3 Space environment?

2.4 Attitude representations

Attitude is not only were one object is located relative to other, but also how it is rotated. It is possible to represent attitude in few ways. Here are described two most commonly used for such case: Euler angels and quaternions. Quaternions are used in all presented estimation methods.

2.4.1 Euler angles

Euler angles were described by Leonard Euler in 1776 [57]. They are used for representing an orientation of an object. To fully understand the orientation between two frames it is necessary to have three parameters: one angle for the rotation around each axis. Those angles are called roll, pitch, yaw typically written as ϕ, θ and ψ respectively. The Euler angles are often used for the definition of rotation matrices about the x, y and z-axis. The equations 1, 2 and 3 describe those rotation matrices.

$$\boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{1}$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2.4.2 Quaternions

In 1843 Sir William Rowan Hamilton described quaterions Hamilton [58] and in 1845 multiplication of quaternions to describe rotations were published by Arthur Cayley Cayley [59]. Quaternions consist of four elements: three give the coordinates and fourth describing angle of rotation Courant and Hilbert [60]. A quaternion can be described as a four-dimensional vector:

$$\boldsymbol{q} \coloneqq \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4}$$

where real part can be described using a rotation angle v, as follows:

$$q_0 = \cos(v/2) \tag{5}$$

$$\boldsymbol{n} = \frac{\boldsymbol{n}}{||\boldsymbol{n}||} \tag{6}$$

The imaginary part can be written as a vector:

$$\boldsymbol{q}_{vec} \coloneqq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = [\boldsymbol{n} \sin(v/2)] \tag{7}$$

where $\boldsymbol{n} = \frac{\boldsymbol{n}}{||\boldsymbol{n}||}$.

2.4.3 Advantages of quaternions

Quaternions are easier to use in calculations, have more compact notation and errors easier to handle than in matrix representation. Also normalization of quaternions is easier and cheaper computationally than normalization of matrices. The most important advantage of quaternions over Euler angles is being safe from gimbal lock. Gimbal lock is the loss of one degree of freedom

that occurs when axes of two of three gimbals are driven into a parallel configuration. This could lead to real disaster Shoemake [63].

Quaternions show many advantages, and due to possible spins of CubeSat, quaternions are the only logical choice to be used for attitude estimation methods.

3 Star-tracker program

Star-tracker program is designed to determine the attitude of the satellite with the image of the stars made by the satellite camera. Before the start of the mission star catalogue is generated, based on the star catalogues obtained from the observatories on Earth and uploaded to the computer on-board the satellite.

After the start, the satellite is released from rocket's tank in space and has no idea where it is. Then the star-tracker on the satellite enters into Lost-In-Space mode (LIS) and analyses the current image of the stars of the camera, and then searches the corresponding result of stars in the on-board star catalogue database. If it finds an entry in the database, the satellite goes into tracking mode. This means each next calculation of the orientation of the satellite attitude is going to be based on a comparison of the current pictures of stars with the preceding ones. If it cannot find the result in the database, this action is repeated from time to time, until a match is found in the database and the program will go into tracking mode.

Of course LIS does not happen only at the beginning of the satellite's flight, but can also result from many causes, e.g. a satellite which is for a long time in the darkness may discharge its battery, and during the next entry into sunlit zone will turn on and look again its attitude. Another case is when satellite becomes lost, although once found the orientation and follow her. This happens, because the successive results are based on preceding ones and even the smallest mistake will grow until the satellite will not be able to correctly calculate their attitude. It may also happen that the satellite will rotate around its axis so fast that the program would not keep up with the processing the image and calculations. In this case, the corresponding other satellite's systems should take an action reducing his spin, but this is not part of this work. It is showed in simple conceptual diagram - Figure 7.

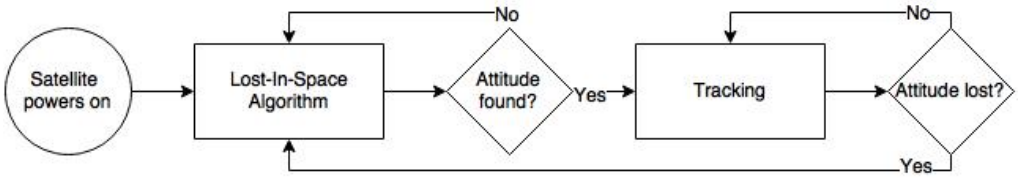


Figure 7: Startracker conceptual algorithm diagram

Each part - LIS and tracking - consists of smaller algorithms. Lost-In-

Space at the beginning of reading an image from the camera, thresholds the image. Thresholding is basically throwing away parts of the image which do not exceed the threshold brightness, making just bright enough stars taken into account. Next it calculates centroid (center of mass) of selected stars, identifies these stars using one of the possible methods (here Planar Triangle, described later) and searches the directory on-board technology (k-vector). If the database does not contain corresponding stars, the algorithm returns to the input state and starts analysing the next image. If it manages to find the corresponding stars, the program determines the attitude of the satellite (by what angle is satellite shifted comparing to the referenced object - Earth or previous image) and enters tracking mode. Tracking mode is in large part similar to the LIS. The algorithm also first analyses the photo, selects and calculates centroids stars, identifies them, but does not search for stars in the directory board. In this algorithm are compared two images, one following after the other, and on this basis the current attitude is calculated.

Generally star-tracker is divided into three main partsMcBryde and Lightsey [65]:

- recognising stars on the image and converting the data into list of star vectors by calculating star centroids;
- identifying which star vector represents which real star in catalogue. This is done by comparing star vectors from the image with data in star catalogue, which is generated before space mission;
- estimating the attitude by calculating the displacement between two frames.

3.1 Centroid - start recognition

Samaan et al. [4]

Liebe [3]

The whole star-tracker program is based on very precise calculations. For this reason, the calculation based on the position of the pixels only can give incorrect results. It is necessary to calculate the position of stars with an

accuracy exceeding pixels. This is why the calculation of the star centroids is necessary.

The first step is the determination of stars' position in the plane of the image. If focused star pictures are recorded, the image of each star will fall only on one or two pixels, and most likely the saturate these pixels, resulting in a pixel level accuracy. Many star-trackers are doing deliberately blurred images, in order to spread the photons on a larger number of pixels, which allows the algorithm for calculating the centroid in sub-pixel accuracy.

After the registration of such image, the centroid of the star is found similarly to the centroid weight points array, with a few differences. Firstly, instead of weight light intensity is used. Secondly, the intensity of light is usually normalized by the pixels around the star in order to filter out glare or noise. The resulting outcome is a series of two-dimensional coordinates on the photo plane with the starting point at the center of the image. This allows the system to the coordinates of the stars can be easily converted into unit vectors in the next step.

The algorithm requires specification of the light intensity threshold (to select the brightest stars) I_{thresh} and the size of the Region of Interest (ROI) a_{ROI} in pixels. These values are adjusted to manipulate the performance of the algorithm. For example, the higher the value of I_{thresh} is more resistant to noise, but can miss some real stars in the picture. Similarly, a large value a_{ROI} means a more exact value of centroid, but the algorithm can see one star there, where are actually two within a short distance of each other. The centroiding part is about trade-off between noise resistance and star recognition, and also between recognizing few close stars as one and recognizing one big star as a few. Please note that a_{ROI} must be a positive odd number for the proper functioning of the algorithm.

The idea of how to calculate such centroids is adapted from McBryde and Lightsey [65] and described below:

1. For each pixel at image coordinates (x,y) with light intensity $I(x, y) > I_{thresh}$, the ROI is defined as the square of pixels with side length a_{ROI} and bottom-left corner at (x_{start}, y_{start})

$$x_{start} = x - \frac{a_{ROI} - 1}{2} \tag{8}$$

$$y_{start} = y - \frac{a_{ROI} - 1}{2} \tag{9}$$

$$x_{end} = x_{start} + a_{ROI} \tag{10}$$

$$y_{end} = y_{start} + a_{ROI} \tag{11}$$

2. If $x_{start} < 0$ or $y_{start} < 0$, discard the pixel and return to previous step with the next pixel.

3. Find the average intensity value of the border pixels I_{border}

$$I_{bottom} = \sum_{i=1}^{x_{end}-1} I(i, y_{start}) \tag{12a}$$

$$I_{top} = \sum_{i=2}^{x_{end}} I(i, y_{end}) \tag{12b}$$

$$I_{left} = \sum_{j=1}^{y_{end}-1} I(x_{start}, j) \tag{12c}$$

$$I_{right} = \sum_{j=2}^{y_{end}} I(x_{start}, j) \tag{12d}$$

$$I_{border} = \frac{I_{top} + I_{bottom} + I_{left} + I_{right}}{4(a_{ROI} - 1)} \tag{12e}$$

4. Normalize all the non-boarder pixels, yielding normalized light intensity matrix $\tilde{I}$

$$\tilde{I}(x, y) = I(x, y) - I_{border} \tag{13}$$

5. Calculate centroid locations (x_{CM}, y_{CM}) and brightness B

$$B = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \tilde{I}(i, j) \tag{14}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{i \times \tilde{I}(i, j)}{B} \tag{15}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{j \times \tilde{I}(i, j)}{B} \tag{16}$$

6. When the centroid location has been calculated for each pixel above the intensity threshold, it is still necessary to make clustering of them. This is done via averaging together all values that are clustered together. Before clustering it is sure that some stars will have more centroids than one, and clustering helps merge them into one. It is configurable by the number of proximity pixels of the centroids. Of course it is not possible to do it perfectly. The main problem is that sometimes there are few stars in close proximity to each other on the photo and it is possible to count those few stars as one, while doing this correctly for all other stars in the photo. On the other hand, if threshold is lowered too much, it is possible to count few stars correctly, but also possible to count one star as a few. The Figure 8 shows example of such trade-off. The clustering can be accomplished by checking each new centroid against a list of already processed centroids. If the new location is within the given range (for example 5 pixels) it is assumed they represent the same star and therefore they values are averaged. The output of this step is the list of star centroids, where each centroid should represent separate star.

7. Now convert each found centroid location into unit vector $\boldsymbol{u}$ using the camera pixel size μ and camera focal length f

$$\boldsymbol{u} = \frac{\begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix}^T}{|| \begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix} ||} \tag{17}$$

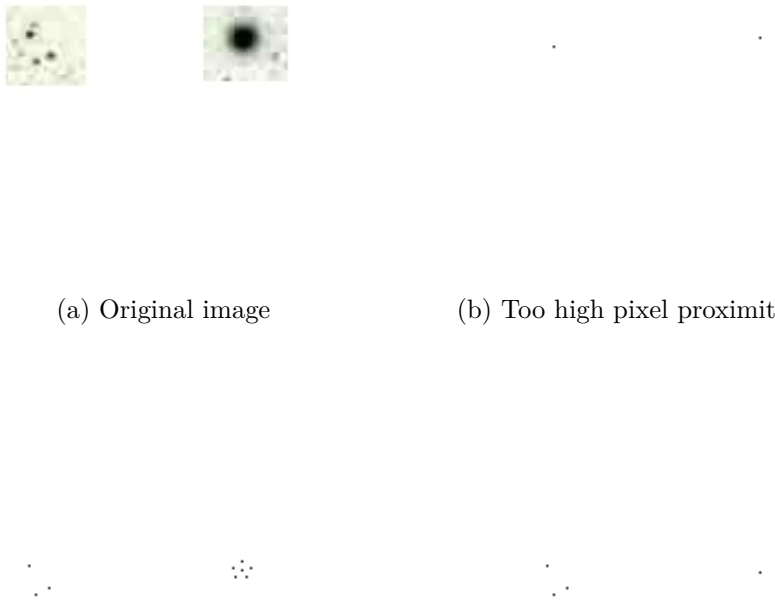
3.2 Star identification

all Spratling and Mortari [66]

Brightness Independent 4-Star Matching Algorithm for Lost-in-Space 3-Axis Attitude AcquisitionDong et al. [11]

SP-Search: A New Algorithm for Star Pattern Recognition Mortari [15]

Star Identification using Neural networks Miri and Shiri [16] Lindblad et al. [17]



(a) Original image

(b) Too high pixel proximity threshold

(c) Too low pixel proximity threshold

(d) Ideal result of clustering

Figure 8: Example of centroids' clustering trade-off (in negative colours).

Star pattern recognition using neural networks Li et al. [18]

3.2.1 Angle Matching

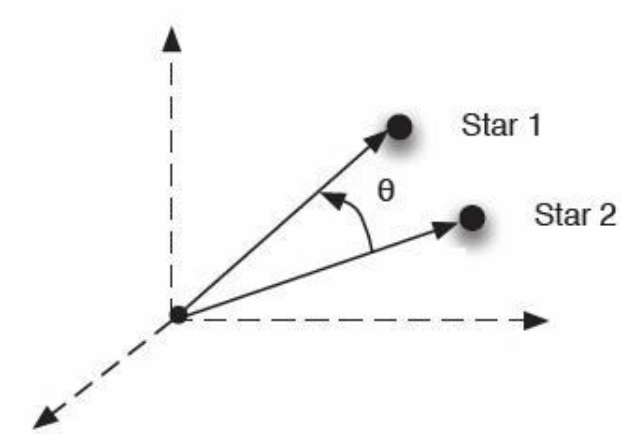


Figure 9: Vector angle method, Image Gottlieb [67]

Gottlieb [67]

$$\theta = \cos^{-1}(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2) \tag{18}$$

$$\boldsymbol{b}_i = A\boldsymbol{r}_i \tag{19}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i, \quad \boldsymbol{v}_i^T A\boldsymbol{r}_i = 0 \tag{20}$$

$$E\{\boldsymbol{v}_i\} = 0 \tag{21a}$$

$$E\left\{\boldsymbol{v}_i\boldsymbol{v}_i^T\right\} = \sigma_i^2[\boldsymbol{I} - (A\boldsymbol{r}_i)(A\boldsymbol{r}_i)^T] \tag{21b}$$

$$\boldsymbol{b}_i^T\boldsymbol{b}_j = \boldsymbol{r}_i^T A^T A\boldsymbol{r}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{22}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i$$
$$\tilde{\boldsymbol{b}}_j = A\boldsymbol{r}_j + \boldsymbol{v}_j$$

$$z \equiv \tilde{\boldsymbol{b}}_i^T\tilde{\boldsymbol{b}}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j + \boldsymbol{r}_i^T A^T\boldsymbol{v}_J + \boldsymbol{r}_j^T A^T\boldsymbol{v}_i + \boldsymbol{v}_i^T\boldsymbol{v}_j \tag{24}$$

$$E\{z\} = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{25}$$

$$p \equiv z - E\{z\} = \boldsymbol{r}_i^T A^T\boldsymbol{v}_J + \boldsymbol{r}_j^T A^T\boldsymbol{v}_i + \boldsymbol{v}_i^T\boldsymbol{v}_j \tag{26}$$

$$\sigma_p^2 \equiv E\{p\} = \boldsymbol{r}_1^T A^T R_2 A\boldsymbol{r}_1 + \boldsymbol{r}_2^T A^T R_a A\boldsymbol{r}_2 + Trace(R_1R_2) = Trace(A\boldsymbol{r}_1\boldsymbol{r}_1^T R_2) + Trace(A\boldsymbol{r}_2\boldsymbol{r}_2^T R_1) + Trace(R_1R_2) \tag{27}$$

3.2.2 Spherical Triangle Matching

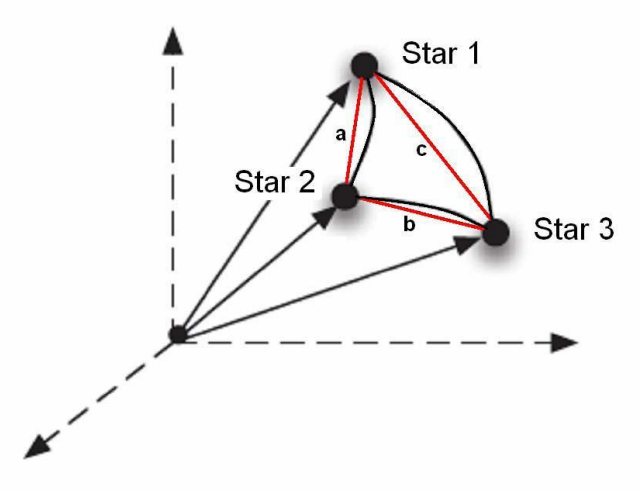


Figure 10: Spherical Triangle Method, Image Cole and Crassidus [14]

Cole and Crassidus [14]

$$A = 4\tan^{-1}\sqrt{\tan\frac{s}{2}\tan\frac{s-a}{2}\tan\frac{s-b}{2}\tan\frac{s-c}{2}} \tag{28}$$

$$s = \frac{1}{2}(a+b+c)$$

$$a = \cos^{-1}\left(\frac{\boldsymbol{b}_1 \cdot \boldsymbol{b}_2}{|\boldsymbol{b}_1||\boldsymbol{b}_2|}\right)$$

$$b = \cos^{-1}\left(\frac{\boldsymbol{b}_2 \cdot \boldsymbol{b}_3}{|\boldsymbol{b}_2||\boldsymbol{b}_3|}\right)$$

$$c = \cos^{-1}\left(\frac{\boldsymbol{b}_3 \cdot \boldsymbol{b}_1}{|\boldsymbol{b}_3||\boldsymbol{b}_1|}\right)$$

$$I_p = \sum \theta^2 dA \tag{30}$$

3.2.3 Planar Triangle

Cole and Crassidis [12]

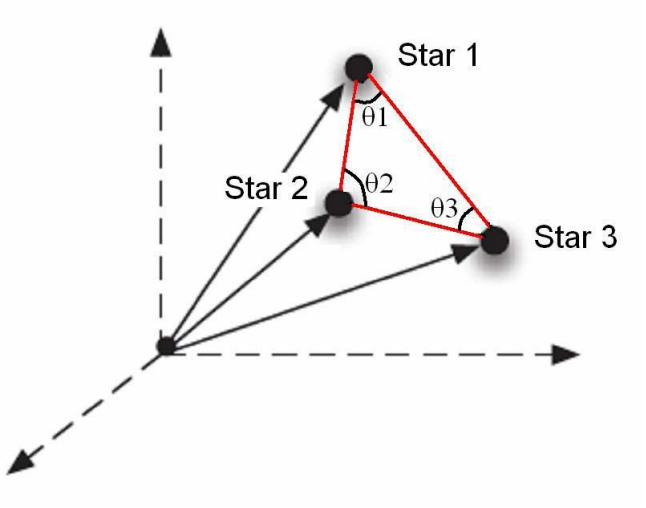


Figure 11: Planar Triangle Method, Image Cole and Crassidis [12]

$$s = \frac{1}{2}(a+b+c) \tag{31a}$$

$$a = ||\boldsymbol{u_p} - \boldsymbol{u_q}|| \tag{31b}$$

$$b = ||\boldsymbol{u_q} - \boldsymbol{u_r}|| \tag{31c}$$

$$c = ||\boldsymbol{u_p} - \boldsymbol{u_r}|| \tag{31d}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{32}$$

$$J = A\frac{(a^2+b^2+c^2)}{36} \tag{33}$$

Derivatives

$$H = \begin{bmatrix} \boldsymbol{h}_1^T & \boldsymbol{h}_2^T & \boldsymbol{h}_3^T \end{bmatrix} \tag{34}$$

$$\boldsymbol{h}_1^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{35a}$$

$$\boldsymbol{h}_2^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{35b}$$

$$\boldsymbol{h}_3^T \equiv \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} \tag{35c}$$

$$\frac{\delta A}{\delta a} = \frac{u_1 - u_2 + u_3 + u_4}{4A} \tag{36a}$$

$$\frac{\delta A}{\delta b} = \frac{u_1 + u_2 - u_3 + u_4}{4A} \tag{36b}$$

$$\frac{\delta A}{\delta c} = \frac{u_1 + u_2 + u_3 - u_4}{4A} \tag{36c}$$

$$u_1 = (s-a)(s-b)(s-c) \tag{37a}$$

$$u_2 = s(s-b)(s-c) \tag{37b}$$

$$u_3 = s(s-a)(s-c) \tag{37c}$$

$$u_4 = s(s-a)(s-b) \tag{37d}$$

$$\frac{\delta a}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_2)^T/a, \quad \frac{\delta a}{\delta \boldsymbol{b}_2} = -\frac{\delta a}{\delta \boldsymbol{b}_1} \tag{38a}$$

$$\frac{\delta b}{\delta \boldsymbol{b}_2} = (\boldsymbol{b}_2 - \boldsymbol{b}_3)^T/b, \quad \frac{\delta b}{\delta \boldsymbol{b}_3} = -\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{38b}$$

$$\frac{\delta c}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_3)^T/c, \quad \frac{\delta c}{\delta \boldsymbol{b}_3} = -\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{38c}$$

$$\sigma_A^2 = HRH^T \tag{39}$$

$$R \equiv \begin{bmatrix} R_1 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & R_2 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & R_3 \end{bmatrix} \tag{40}$$

Polar Moment

$$\bar{H} = \begin{bmatrix} \bar{\boldsymbol{h}}_1^T & \bar{\boldsymbol{h}}_2^T & \bar{\boldsymbol{h}}_3^T \end{bmatrix} \tag{41}$$

$$\bar{\boldsymbol{h}}_1^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta A}\boldsymbol{h}_1^T \tag{42a}$$

$$\bar{\boldsymbol{h}}_2^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta A}\boldsymbol{h}_2^T \tag{42b}$$

$$\bar{\boldsymbol{h}}_3^T \equiv \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta A}\boldsymbol{h}_3^T \tag{42c}$$

$$\frac{\delta J}{\delta a} = Aa/18, \quad \frac{\delta J}{\delta a} = Ab/18, \quad \frac{\delta J}{\delta a} = Ac/18 \tag{43a}$$

$$\frac{\delta J}{\delta A} = (a^2 + b^2 + c^2)/36 \tag{43b}$$

$$\sigma_J^2 = \bar{H} R \bar{H}^T \tag{44}$$

3.2.4 Pyramid

Mortari et al. [13]

3.2.5 Rate Matching

Samaan et al. [68] to be removed?

3.2.6 Voting

Kolomenkin et al. [9]

3.2.7 Grid

Padgett and Kreutz-Delgado [10]

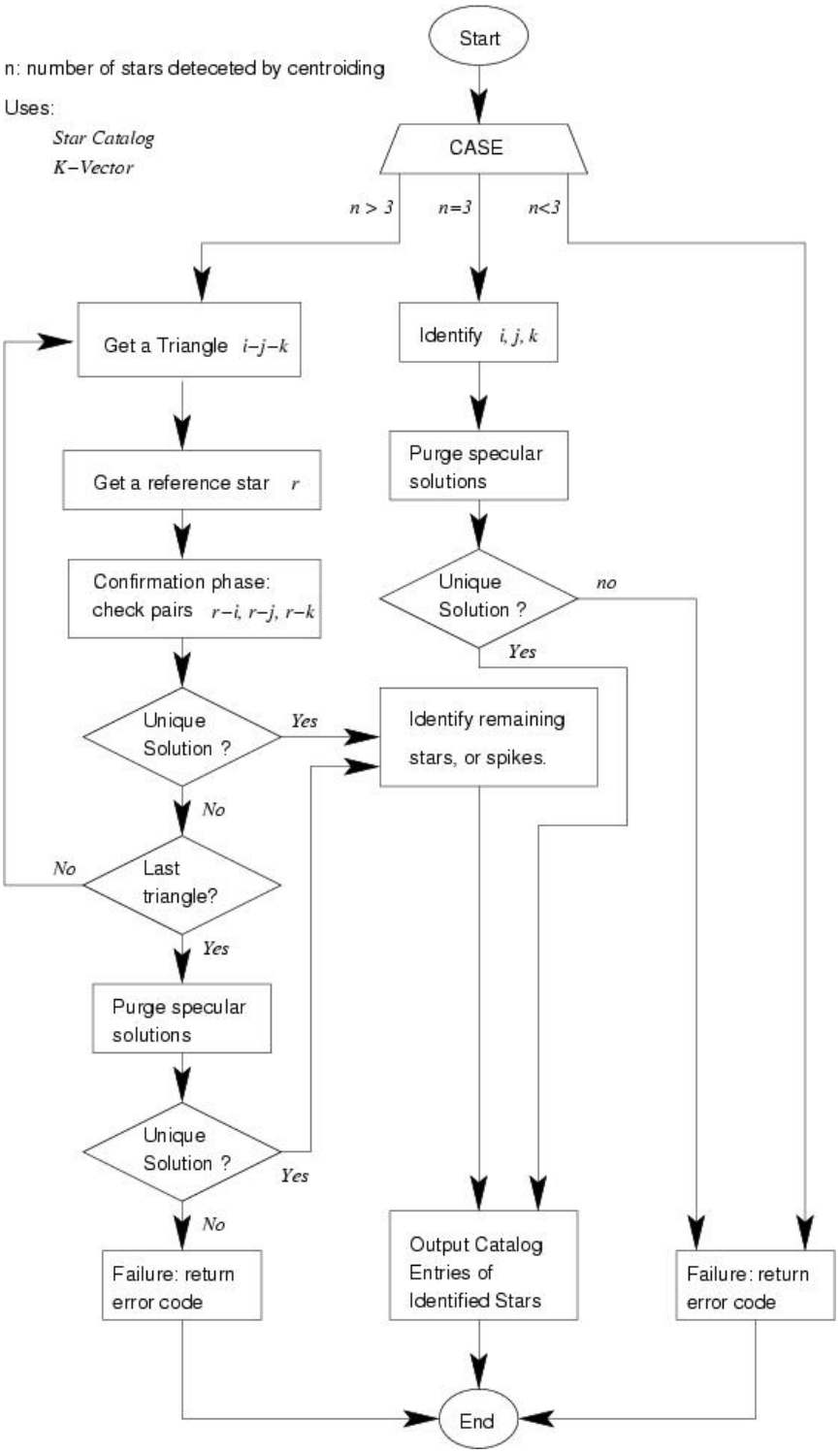


Figure 12: Pyramid Method Flowchart, Image Mortari et al. [13]

3.3 Star-catalogue and searching for matching stars

3.3.1 Star Catalogue Generation

McBryde and Lightsey [65]

$$\boldsymbol{u} = \begin{bmatrix} \cos\alpha\cos\delta \\ \sin\alpha\cos\delta \\ \sin\delta \end{bmatrix} \tag{45}$$

$$m_i \leq m_{max} \tag{46}$$

$$m_j \leq m_{max} \tag{47}$$

$$\boldsymbol{u_a^T u_b} \geq \cos\theta_{FOV} \tag{48}$$

3.3.2 Candidate Matching

to be removed?

3.3.3 Result Verification

to be removed?

3.3.4 Search Less Algorithm and k-vector

Search Less Algorithm (SLA) citetmortari2000k is special algorithm used to speed up looking up the corresponding stars in the on-board computer. Its heart is k-vector technique. Comparing to binary search technique, k-vector demonstrates high speed gain rate, from 10 to more than 50 times [20].

The k-vector database is built before spacecraft's start. The k-vector table is a structural database of all catalogued star pairs that could possibly fit into the camera FOV over the whole sky. The star pairs are ordered with increasing inter-star angle [19].

The k-vector technique works in the following way: $\boldsymbol{y}$ is the data vector of n elements ($n \gg 1$) and $\boldsymbol{s}$ is the same vector but sorted in ascending mode: $\boldsymbol{s}(i) \leq \boldsymbol{s}(i+1)$, and $\boldsymbol{s}(i) = \boldsymbol{y}(\boldsymbol{I}(i))$, where $\boldsymbol{I}$ is the integer vector of the sorting with length n [21].

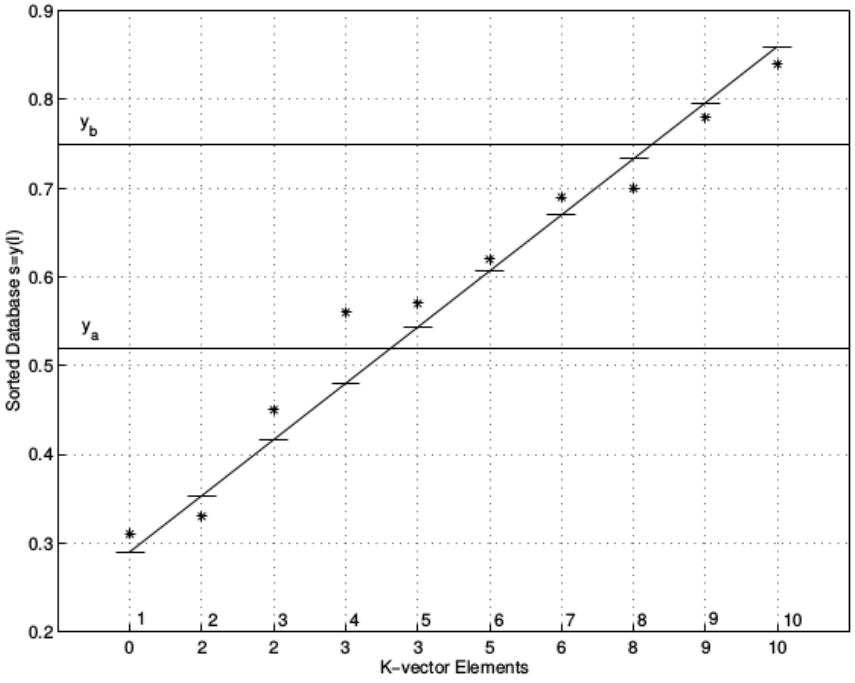


Figure 13: K-vector technique example, Image from [20]

In the Figure 13 it is shown how such k-vector construction looks like. In this example the database has 10 elements. X-axis describes k-vector elements $\boldsymbol{k}(i)$ and y-axis describes the sorted values $\boldsymbol{s}(i)$ (inter-star angle in this case). The value of $\boldsymbol{k}(i)$ represents the number of elements of the database vector $\boldsymbol{y}$ below the value $z(i)$. The resulting k-vector is $\boldsymbol{k} = \{0, 2, 2, 3, 3, 5, 6, 8, 9, 10\}^T$.

The following equations describe mathematically how the k-vector database is created. The line connecting two extreme points, $[1, y_{min}]$ and $[n, y_{max}]$, has to be a little stepper and connect points $[1, y_{min} - \delta\epsilon]$ and $[n, y_{max} + \delta\epsilon]$. This assures $\boldsymbol{k}(1) = 0$ and $\boldsymbol{k}(n) = n$ and simplifies the code by avoiding many index checks.

$$z(x) = mx + q \begin{cases} m = \frac{y_{max} - y_{min} + \delta\epsilon}{n-1} \\ q = y_{min} - m - \delta\epsilon \end{cases} \tag{49}$$

$$\delta\epsilon = (n-1)\epsilon \tag{50}$$

ϵ is relative machine precision for double precision arithmetic

$$\epsilon \approx 22.2 \times 10^{-16} \tag{51}$$

Each other element can be found using the following equation:

$$\boldsymbol{k}(i) = j \quad where \quad s(j) \leq z(i) < s(j+1) \tag{52}$$

or

$$\boldsymbol{k}(i) = j \quad \text{where j is the greatest index such} \quad s(j) \leq \boldsymbol{y}(\boldsymbol{I}(i)) \quad \text{is satisfied.} \tag{53}$$

At this point k-vector has been built, and further use is quite simple. The equation to find indices identifying in $\boldsymbol{s}$ vector all possible data within the range $[y_a, y_b]$ is following:

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \quad and \quad j_t = \left\lceil \frac{y_b - q}{m} \right\rceil \tag{54}$$

In the example of Figure 13 it is $j_b = 4$ and $j_t = 9$. After indices are evaluated it is possible to calculate

$$k_{start} = \boldsymbol{k}(j_b) + 1 \quad and \quad k_{end} = \boldsymbol{k}(j_t) \tag{55}$$

Finally, having found k_{start} and k_{end} it is immediately known, that searched elements are $\boldsymbol{y}(i) \in [y_a, y_b]$, which are all the $\boldsymbol{y}(\boldsymbol{I}(i))$ where k ranges from k_{start} and k_{end}.

Note that in example the searched elements should be $k_{start} = 4$ and $k_{end} = 8$ while the presented technique returns $k_{start} = 4$ and $k_{end} = 9$. This happens due to fact that k_{start} and k_{end} have 50% possibility to not belong to the $[y_a, z(j_a + 1)]$ and $[z(j_b), y_b)]$ ranges respectively.

3.4 Attitude Determination

Jenssen et al. [69]

AIM (Attitude estimation using Image Matching)Delabie [24]

all Hall [31] Markley and Mortari [30]

3.4.1 q-method

$$\boldsymbol{s}_b = \boldsymbol{R}^{bi}\boldsymbol{s}_i \quad \boldsymbol{m}_b = \boldsymbol{R}^{bi}\boldsymbol{m}_i \tag{56}$$

$$\begin{aligned} J &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki})^T(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \\ &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb}^T\boldsymbol{v}_{kb} + \boldsymbol{v}_{ki}^T\boldsymbol{v}_{ki} + 2\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \end{aligned} \tag{57}$$

$$J = \sum w_k(1 - \boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \tag{58}$$

$$g(\boldsymbol{R}) = \sum w_k\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki} \tag{59}$$

$$\boldsymbol{R} = (q_4^2 - \boldsymbol{q}^T\boldsymbol{q})1 + 2\boldsymbol{q}\boldsymbol{q}^T - 2q_4\boldsymbol{q}^{\boldsymbol{x}} \tag{60}$$

$$\bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}} = 1 \tag{61}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T\boldsymbol{K}\bar{\boldsymbol{q}} \tag{62}$$

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{S} - \sigma\boldsymbol{I} & \boldsymbol{Z} \\ \boldsymbol{Z}^T & \sigma \end{bmatrix} \tag{63}$$

$$\boldsymbol{B} = \sum_{k=1}^{N} w_k(\boldsymbol{v}_{kb}\boldsymbol{v}_{ki}^T) \tag{64}$$

$$\boldsymbol{S} = \boldsymbol{B} + \boldsymbol{B}^T \tag{65}$$

$$\boldsymbol{Z} = \begin{bmatrix} B_{23} - B_{32} & B_{32} - B_{13} & B_{12} - B_{21} \end{bmatrix}^T \tag{66}$$

$$\sigma = tr[\boldsymbol{B}] \tag{67}$$

$$g'(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} - \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} \tag{68}$$

$$\boldsymbol{K}\bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}} \tag{69}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} = \bar{\boldsymbol{q}}^T \lambda \bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} = \lambda \tag{70}$$

3.4.2 Wahba's problem

The developed extended QUEST method described in this report builds upon the principles of Wahba's problem. The problem was first stated by Grace Wahba in 1965 Wahba [70]. Given two sets of vector observations, a rotation matrix $\boldsymbol{M}$ can be found which minimizes the orientation error. This is an optimization problem, where the cost function is:

$$\sum_{j}^{n} ||\boldsymbol{r}_j - \boldsymbol{M}\boldsymbol{b}_j|| \tag{71}$$

For satellite attitude determination, the vectors $\boldsymbol{r}_j$ for $j \in \{1, n\}$ are the reference sensor data given in the NED frame. The vectors $\boldsymbol{b}_j$ for $j \in \{1, n\}$ are the measured sensor data in the BODY frame. $\boldsymbol{M}$ is the least squares estimate of the rotation matrix which carries the known frame of reference into the satellite fixed frame of reference. The QUEST method uses this problem in order to minimize the attitude estimation error.

3.4.3 QUEST

improvement to quest implementation Cheng and Shuster [26]

kallman filtering Shuster [25]

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j)^T(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j^T\boldsymbol{b}_j - 2\boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j + \boldsymbol{r}_j^T\boldsymbol{r}_j) \tag{72}$$

$$J(\boldsymbol{q}) = \sum_{j=1}^{n}\frac{1}{\sigma_j^2}(1 - \boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) \tag{73}$$

3.4.4 TRIAD

a must

3.4.5 The Fast Optimal Attitude Matrix

to be removed?

3.4.6 DCM (Direction Cosine Matrix) - Singular Value Decomposition?

Juang et al. [29] and

McBryde and Lightsey [65]

$$\boldsymbol{B} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{b_i r_i^T} \tag{74}$$

$$\boldsymbol{B} = \boldsymbol{USV^T} \tag{75}$$

$$\boldsymbol{U}_+ = \boldsymbol{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{U} \end{bmatrix} \tag{76}$$

$$\boldsymbol{V}_+ = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{V} \end{bmatrix} \tag{77}$$

$$\boldsymbol{A} = \boldsymbol{U}_+ \boldsymbol{V}_+^{\boldsymbol{T}} \tag{78}$$

4 Prototype

For now the following parts are finished in Python:

1. Centroiding
2. Planar Triangle Recognition with variations (nearly - without equation 44)
3. Pyramid alg ?
4. k-vector
5. QUEST (not started yet)

Testing

Kruijff et al. [71], Kruijff and vd Heiden [72], Dzamba and Enright [73],

5 Complete program

6 Testing of star-tracker

Tappe et al. [74]

References

[1] M. A. Swartwout, "A brief history of rideshares (and attack of the CubeSats)," in *Aerospace Conference, 2011 IEEE*. IEEE, 2011, pp. 1–15.

[2] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," 2000.

[3] C. C. Liebe, "Accuracy performance of star trackers-a tutorial," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 587–599, 2002.

[4] M. A. Samaan, D. Mortari, T. Pollock, and J. L. Junkins, "Predictive centroiding for single and multiple FOVs star trackers," *Advances in the Astronautical Sciences*, vol. 112, pp. 59–71, 2002.

[5] M. W. Knutson, "Fast star tracker centroid algorithm for high performance CubeSat with air bearing validation," Master's thesis, Massachusetts Institute of Technology, 2012.

[6] M. Azizabadi, A. Behrad, and M. Ghaznavi-Ghoushchi, "VLSI implementation of star detection and centroid calculation algorithms for star tracking applications," *Journal of real-time image processing*, vol. 9, no. 1, pp. 127–140, 2014.

[7] M. Lindh, "Development and implementation of star tracker electronics," Master's thesis, 2014.

[8] P. Zhang, Q. Zhao, J. Liu, and N. Liu, "A brightness-referenced star identification algorithm for aps star trackers," *Sensors*, vol. 14, no. 10, pp. 18 498–18 514, 2014.

[9] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 441–456, 2008.

[10] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 202–213, 1997.

[11] Y. Dong, F. Xing, and Z. You, "Brightness independent 4-star matching algorithm for lost-in-space 3-axis attitude acquisition," *Tsinghua Science & Technology*, vol. 11, no. 5, pp. 543–548, 2006.

[12] C. L. Cole and J. L. Crassidis, "Fast star-pattern recognition using planar triangles," *Journal of guidance, control, and dynamics*, vol. 29, no. 1, pp. 64–71, 2006.

[13] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.

[14] C. L. Cole and J. Crassidus, "Fast star pattern recognition using spherical triangles," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit. Providence, Rhode Island: AIAA*, 2004.

[15] D. Mortari, "SP-search: A new algorithm for star pattern recognition," *Advances in the Astronautical Sciences*, vol. 102, no. Pt II, pp. 1165–1174, 1999.

[16] S. S. Miri and M. E. Shiri, "Star identification using Delaunay triangulation and distributed neural networks," *International Journal of Modeling and Optimization*, vol. 2, no. 3, p. 234, 2012.

[17] T. Lindblad, C. S. Lindsey, Å. Eide, Ö. Solberg, and A. Bolseth, "Star Identification using Neural Networks," 1997.

[18] C. Li, K. Li, L. Zhang, S. Jin, and J. Zu, "Star pattern recognition method based on neural network," *Chinese Science Bulletin*, vol. 48, no. 18, pp. 1927–1930, 2003.

[19] D. Mortari, "A fast on-board autonomous attitude determination system based on a new star-ID technique for a wide FOV star tracker," *Advances in the Astronautical Sciences*, vol. 93, pp. 893–904, 1996.

[20] D. Mortari and B. Neta, "K-vector range searching techniques," *Adv. Astronaut. Sci*, vol. 105, pp. 449–464, 2000.

[21] D. Mortari and J. Rogers, "A k-vector Approach to Sampling, Interpolation, and Approximation," *The Journal of the Astronautical Sciences*, vol. 60, no. 3-4, pp. 686–706, 2013.

[22] J. A. Tappe, "Development of star tracker system for accurate estimation of spacecraft attitude," Master's thesis, Monterey, California. Naval Postgraduate School, 2009.

[23] L. Feruglio, S. Corpino, and D. Calvi, "Neural networks for event detection: an interplanetary cubesat asteroid mission case study," in *AIAA SPACE 2016*, 2016, p. 5615.

[24] T. Delabie, "A highly efficient attitude estimation algorithm for star trackers based on optimal image matching," in *AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota*, 2012.

[25] M. Shuster, "Kalman filtering of spacecraft attitude and the QUEST model," *Journal of the Astronautical Sciences*, vol. 38, pp. 377–393, 1990.

[26] Y. Cheng and M. D. Shuster, "Improvement to the Implementation of the QUEST Algorithm," *Journal of Guidance, Control, and Dynamics*, 2014.

[27] M. Psiaki, "Extended quest attitude determination filtering," in *NASA CONFERENCE PUBLICATION*. NASA, 1999, pp. 1–16.

[28] T. B. Rinnan, "Development and comparison of estimation methods for attitude determination," Master's thesis, Institutt for teknisk kybernetikk, 2012.

[29] J.-N. Juang, H.-Y. Kim, and J. L. Junkins, "An efficient and robust singular value method for star pattern recognition and attitude determination," *NASA, Tech. Rep. TM-2003-212142*.

[30] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," 1999.

[31] C. D. Hall, "Spacecraft attitude dynamics and control," *Lecture Notes posted on Handouts page [online]*, vol. 12, no. 2003, 2003. [Online]. Available: http://www.dept.aoe.vt.edu/ cdhall/courses/aoe4140/attde.pdf

[32] M. Gąska, "Moduł komputera pokładowego do satelity CubeSat z funkcją Star Tracker," Master's thesis, Warsaw University of Technology, 2016.

[33] D. Felikson, J. Hahmall, M. F. Vess, and M. Ekinci, "On-Orbit Solar Dynamics Observatory (SDO) Star Tracker Warm Pixel Analysis," in *AIAA Guidance, Navigation and Control Conference, Portland, Oregon*, vol. 6728, 2011.

[34] R. Kandiyil, "Attitude determination software for a star sensor," Master's thesis, Luleå University of Technology, 2010.

[35] K. M. Huffman *et al.*, "Designing star trackers to meet micro-satellite requirements," Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2006.

[36] K. D. Diaz, "Performance Analysis Of A Fixed Point Star Tracker Algorithm," Master's thesis, California Polytechnic State University, 2006.

[37] E. Jalabert, E. Fabacher, N. Guy, S. Lizy-Destrez, W. Rappin, and G. Rivier, "Optimization of star research algorithm for ESMO star tracker," 2011.

[38] S. Lizy-Destrez and D. Mimoun, "Str: a student developed star tracker for the esa-led esmo moon mission," 2010.

[39] A. Rose, "STAR integrated tracker," *arXiv preprint nucl-ex/0307015*, 2003.

[40] D. Mortari and A. Romoli, "StarNav III: a three fields of view star tracker," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 1–57.

[41] M. Cannata, M. Greene, S. Mulligan, and V. Popovici, "Autonomous star-imaging attitude sensor (ASIAS)," *Faculty of Science and Engineering. York University*, pp. 2006–07, 2007.

[42] M. Swartwout, "The first one hundred cubesats: A statistical look," *Journal of Small Satellites*, vol. 2, no. 2, pp. 213–233, 2013.

[43] "Nanosatsatellite and CubeSat Database," http://www.nanosats.eu/, accessed: 2017-08-15.

[44] "PW-Sat2: oficjalna strona projektu PW-SAT 2 (oraz PW-SAT)," https://pw-sat.pl/, accessed: 2017-08-15.

[45] "BRITE-PL Pierwszy polski satelita naukowy," http://www.brite-pl.pl/pliki/nauka.html, accessed: 2017-08-15.

[46] "Polskie Radio: Pierwszy polski satelita poleciał w kosmos," http://www.polskieradio.pl/7/129/Artykul/536883,Pierwszy-polski-satelita-polecial-w-kosmos, accessed: 2017-08-15.

[47] "BRITE-PL, PL2 (CanX 3C, 3D / Lem, Heweliusz) - Gunter's Space Page," http://space.skyrocket.de/doc_sdat/brite-pl.htm, accessed: 2017-08-15.

[48] "Future Planetary Expeditions: CubeSats to the Planets," http://futureplanets.blogspot.com/2013/10/, accessed: 2017-08-15.

[49] "Warsaw University of Technology - News - PW-SAT 2 gets funding," https://www.pw.edu.pl/Studenci/Aktualnosci/Satelita-studencki-PW-Sat2-otrzyma-dofinansowanie, accessed: 2017-08-15.

[50] "GPS: The Global Positioning System," http://www.gps.gov/systems/gps/space/, accessed: 2017-08-18.

[51] "NASA's Utilization of Global Positioning System (GPS)," https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS_Utilization.html accessed: 2017-08-18.

[52] "Space Station Using GPS in Attitude Control," https://www.nasa.gov/centers/johnson/news/releases/2002/j02-61.html, accessed: 2017-08-18.

[53] V. Capuano, C. Botteron, Y. Wang, J. Tian, J. Leclère, and P.-A. Farine, "GNSS/INS/Star tracker integrated navigation system for Earth-Moon transfer orbit," in *ION GNSS+ 2014*, no. EPFL-CONF-202129, 2014.

[54] W. J. Larson and J. R. Wertz, "Space mission analysis and design," Microcosm, Inc., Torrance, CA (US), Tech. Rep., 1992.

[55] "Earth's navigation orbits image," https://upload.wikimedia.org/wikipedia/commons/thumb Comparison_satellite_navigation_orbits.svg.png, accessed: 2017-08-18.

[56] "NED frame image," http://www.basicairdata.eu/knowledge-center/background-topics/coordinate-system/, accessed: 2017-09-02.

[57] L. Euler, "Formulae generales pro translatione quacunque corporum rigidorum," *Novi Acad. Sci. Petrop*, vol. 20, pp. 189–207, 1775.

[58] W. R. Hamilton, "LXXVIII. On quaternions; or on a new system of imaginaries in Algebra: To the editors of the Philosophical Magazine and Journal," *Philosophical Magazine Series 3*, vol. 25, no. 169, pp. 489–495, 1844.

[59] A. Cayley, "XIII. On certain results relating to quaternions: To the editors of the Philosophical Magazine and Journal," *Philosophical Magazine Series 3*, vol. 26, no. 171, pp. 141–145, 1845.

[60] R. Courant and D. Hilbert, "Methods of mathematical physics, Volume I," 1953.

[61] J. E. Mebius, "A matrix-based proof of the quaternion representation theorem for four-dimensional rotations," *arXiv preprint math/0501249*, 2005.

[62] M. Barile, "Conjugate elements," 1997. [Online]. Available: http://mathworld.wolfram.com/ConjugateElements.html

[63] K. Shoemake, "Animating rotation with quaternion curves," in *ACM SIGGRAPH computer graphics*, vol. 19, no. 3. ACM, 1985, pp. 245–254.

[64] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[65] C. R. McBryde and E. G. Lightsey, "A star tracker design for CubeSats," in *Aerospace Conference, 2012 IEEE*. IEEE, 2012, pp. 1–14.

[66] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, 2009.

[67] D. Gottlieb, "Star pattern recognition techniques," *Spacecraft Attitude Determination and Control, The Netherlands*, pp. 257–266, 1978.

[68] M. A. Samaan, D. Mortari, and J. L. Junkins, "Recursive mode star identification algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1246–1254, 2005.

[69] K. L. Jenssen, K. H. Yabar, and J. T. Gravdahl, "A comparison of attitude determination methods: theory and experiments," in *proceedings of the 62nd International Astronautical Congress, Cape Town, South Africa*, 2011, pp. 3–7.

[70] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[71] M. Kruijff, E. Heide, C. De Boom, and N. Heiden, "Star sensor algorithm application and spin-off," in *54th International Astronautical Congress of the International Astronautical Federation(IAF)*, 2003.

[72] M. Kruijff and N. vd Heiden, "Automated star sensor performance assessment using real-sky data of MEFIST II," *Moon*, vol. 100, no. 1, pp. 500–2, 2003.

[73] T. Dzamba and J. Enright, "Ground testing strategies for verifying the slew rate tolerance of star trackers," *Sensors*, vol. 14, no. 3, pp. 3939–3964, 2014.

[74] J. Tappe, J. J. Kim, A. Jordan, and B. Agrawal, "Star tracker attitude estimation for an indoor ground-based spacecraft simulator," in *AIAA Guidance, Navigation, and Control Conference*, vol. 1, 2011, pp. 1116–1122.

List of Tables

List of Figures